

Puska helyett

Jelige: SZG 012

$$\sum_{w \in E} f(w) = \sum_{v \in E} f(v)$$

$$\theta(g(n)) = \Omega(g(n)) \cap O(g(n))$$

Puska helyett

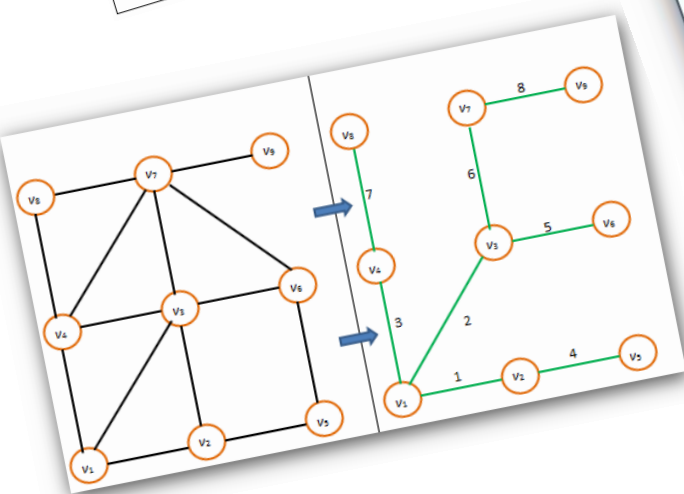
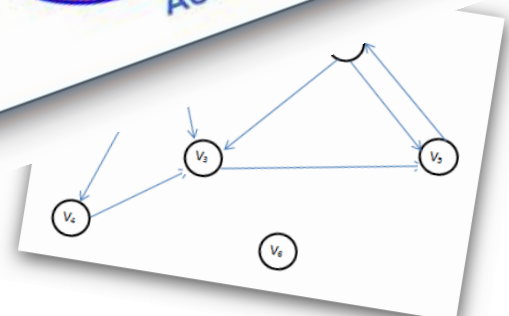
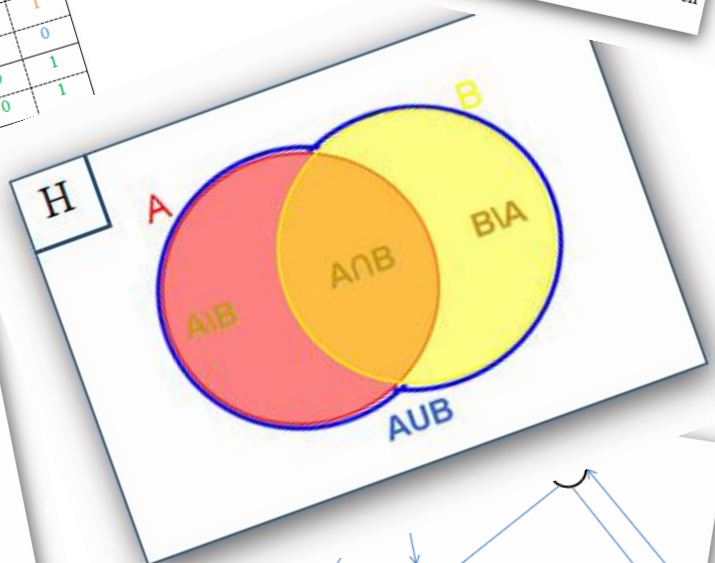
Csak kétféleképpen élheted az életed. Vagy abban hiszel, a világon semmi sem varázslat. Vagy pedig abban, hogy a világon minden varázslat.

(Albert Einstein)

Lássuk be, nem egy nagy feladat ezt bizonyítani, de a kedvéért azért kísérletet teszünk rá.

A	1	1	1	1	0	1	0	1	0
B	1	1	0	1	0	1	0	1	0
C	0	0	1	0	1	0	1	0	1
$\bar{A}$	0	0	0	0	1	0	0	0	1
$\bar{B}$	0	1	0	0	0	0	1	0	0
$\bar{C}$	1	1	0	0	0	1	0	0	0
$A \wedge B$	0	1	0	0	0	0	1	0	0
$A \wedge B \wedge \bar{C}$	0	0	0	0	0	0	0	0	1
$\bar{A} \wedge B$	0	0	0	0	0	0	0	0	0
$\bar{A} \wedge B \wedge \bar{C}$	0	0	0	0	0	0	0	0	0
$A \wedge \bar{B}$	0	0	0	0	0	0	0	1	0
$A \wedge \bar{B} \wedge \bar{C}$	0	1	0	0	0	0	0	0	0
$(A \wedge B \wedge \bar{C}) \vee (\bar{A} \wedge B \wedge \bar{C})$	0	1	0	0	0	0	0	0	1
$(A \wedge B \wedge \bar{C}) \vee (\bar{A} \wedge B \wedge \bar{C}) \vee (\bar{A} \wedge \bar{B} \wedge \bar{C})$	0	1	0	0	0	0	0	0	1
$\bar{A} \wedge B \wedge C$	0	0	1	0	0	0	0	0	0

$A_1 \times \dots \times A_n := \{(a_1, \dots, a_n) \mid a_i \in A_i, \forall i = 1, \dots, n\}$
 Azt jelenti, hogy a halmazok úgynevezett Descartes-féle szorzata egy olyan halmaz, melynek a tagjai az $a_1, a_2, a_3, \dots, a_n$ -el jelölt elemek, ahol is a_i (i -edik halmazból származó tag) az A_i (i -edik halmaz) eleme, és az i -k 1-től mennek egyesével n -ig (n elméletben lehet akár 17563893 is).



Tartalom

I. Bevezető, Előszó helyett	5
II. Motiváció	7
III. Becslések	9
<i>Példák</i>	10
IV. Igazságtáblák	13
<i>Diszjunktív normálforma</i>	15
V. Halmazelmélet, Rendezések, Relációk	17
<i>Homogén bináris relációk</i>	19
<i>Példa feladat</i>	20
VI. Gráfelmélet és Gráfalgoritmusok	23
<i>Gráfokkal kapcsolatos alapfogalmak</i>	24
<i>Szélességi bejárás</i>	26
<i>Mélylési bejárás</i>	27
<i>Kruskal algoritmusa</i>	28
<i>Dijkstra algoritmusa</i>	29
<i>A Gale-Shapley-Algoritmus</i>	31
<i>Hálózati folyamatok</i>	31
VII. Rendezések	34
<i>Példa</i>	34
<i>Beszűrő rendezéssel</i>	34
<i>Összefésülő rendezéssel</i>	35
<i>Kupacrendezéssel</i>	36
<i>Gyorsrendezéssel</i>	39
<i>Radix rendezés</i>	40
VIII. Formális nyelvek és automaták	41
<i>Grammatika</i>	41
<i>Chomsky féle nyelvosztályok</i>	42
➤ 0. típusú nyelvtanok (rekurzíve felsorolható nyelvtanok)	42
➤ 1. típusú nyelvtanok (környezetfüggő nyelvtanok)	42
➤ 2. típusú nyelvtanok (környezetfüggetlen nyelvtanok)	42
➤ 3. típusú nyelvtanok (reguláris nyelvtanok)	43
Véges determinisztikus automaták	43
<i>Példa feladat</i>	44

IX. Utószó helyett	46
Melléklet	
X. Feladatmegoldó Program	49
XI. Példatár - Feladatok	50
Becslések	50
Halmazelmélet, igazságtáblák és relációk	50
Gráfelmélet és Gráfalgoritmusok	51
Rendezések.....	58
Formális nyelvek és automaták	58
XII. Példatár - Megoldások	60
Becslések	60
Halmazelmélet, Rendezések, Relációk.....	62
Gráfelmélet és Gráfalgoritmusok	68
Rendező algoritmusok.....	75
Formális nyelvek és automaták	83
XIII. Irodalomjegyzék és ajánlott irodalom	85
XIV. Algoritmusok.....	87
XV. Tanári vélemények	90
XVI. Az általunk tanult Számítástudomány tantárgyi tematikája, BGF-GKZ	92

I. Bevezető, Előszó helyett

„Sok dolog van, ami lehetetlennek látszik, mindaddig, amíg valaki meg nem kísérli azt.”

(André Gide)

Hol is kezdjük...? ☺

Először is, Üdvözöllek Kedves Hallgató-Olvasó!

Remélem, nem veszed zokon, ha a tantárgyi részeknél szakítunk a megszokott megszólítással és hangnemmel. Ez a dokumentum tulajdonképpen egy segédlet a BGF-GKZ-n oktatott Számítástudomány nevű tantárgyhoz. Nektek hallgatóknak készült azért, hogy egy helyen legyen a tantárgyhoz kapcsolódó minél több, hasznos ismeret, lehetőség szerint a legegyszerűbben leírva.

Amikor 2011-ben az évfolyamommal a felsőoktatásba kerültünk, nagy nehézséget jelentett megtanulni a különböző tantárgyak elsajátításához szükséges anyagot, különös nehézséget jelentettek a matematika alapú tantárgyak. A tantárgyak közül a legnehezebbnek a Számítástudomány bizonyult. A tantárgy célja a következő: „Megismertetni a hallgatókkal a számítási folyamatok matematikai vonatkozásait, áttekinteni az informatikában alkalmazott legfontosabb matematikai eszközöket. Bemutatni az algoritmuselmélet és a gráfelméleti alapjait. Bevezetni a hallgatókat az automaták és formális nyelvek elméletébe. Ismertetni a kiszámíthatóság és bonyolultság elméletének elemeit.” (1. melléklet: BGF-GKZ, Tantárgyi program, 2012/2013 tanév, I. félév)

A legnagyobb problémát az okozta, hogy kevés olyan feladat volt elérhető számunkra, ami szorosan ehhez a témakörhöz kapcsolódik, és részletesen el van magyarázva. Ezért született ez a segédanyag, hogy legyen egy dokumentum, ami a számunkra szükséges, alapvető témákról szól a tantárgy keretein belül. Törekedtünk minél egyszerűbb, közérthetőbb módon leírni a nyers elméletet, hozzájuk példákat, kidolgozott feladatokat csatolni, ezzel hozzájárulni a könnyebb megértéshez. Ha komolyabban akarsz foglalkozni a témával, a függelékben megtalálhatod a felhasznált szakirodalmak egész tárházát.

Amikor készülőfélben volt a dokumentum, felvetődött a kérdés, hogy milyen anyagrészeket ragadjunk ki a tantárgyból? Megkérdeztük csoporttársainkat is, és egybehangzóan azon a véleményen voltunk, hogy olyan típusú feladatokra koncentráljunk, amelyek eddig szerepelt a zárthelyi dolgozatokban és a minta dolgozatokban szerepeltek. Legyen benne viszonylag sok feladat, kidolgozva, részletesen elmagyarázott megoldásokkal, valamint a feladatokhoz szükséges, *nélkülözhetetlen* elméleti anyagot is tartalmazza. Tehát a

Puska helyett című művünk nem fedi le teljesen a tananyagot, viszont mindenképpen nagy segítség lehet a felkészülésben. Emellett egy kicsit az is célunk volt, hogy öntsük formába mindazt, amit megtanultunk, hogy ne felejtsük el, legyen mihez visszanyúlni a későbbiekben.

Fontos tudnod, hogy segédanyagunk – melynek írása során több forrásra is támaszkodtunk - nem helyettesítheti az órai prezentációkat, jegyzeteket, tankönyveket, nem mentesít a gyakorlatra és előadásra járás alól, ez csupán egy segédlet. ☺
A felhasznált források mindegyike meg van jelölve a könyv végén, akkor is, ha átvettünk belőle, vagy abból merítettük az elméleti tudást.

Reméljük, hogy egyre többen fogják használni segédanyagunkat és idővel beépülhet az elektronikus segédanyagok közé is. Ha esetleg valaki ki szeretné nyomtatni, akkor érdemes színesben, mivel vannak olyan oldalak, amelyeken a magyarázatok leírásában színeket használunk.

Összefoglalva, célnak a következőt fogalmaztuk meg: készíteni egy kiskönyvet, ami részben lefedi a tananyagot, könnyed nyelvezettel rendelkezik, gyakorlati feladat orientált, hogy a hallgatók könnyebben tudhassák maguk mögött a sikeres vizsgát. Eddigi tapasztalatok alapján a cél megvalósulni látszik.

A könyvben található definíciók és tételek a következő irodalmakból származnak:

- Katona-Recski-Szabó: A Számítástudomány alapjai, Typotex Kiadó, 2006
- Tasnádi Attila: Számítástudomány gazdaságinformatikusoknak Bologna – Tankönyvsorozat, Budapest 2006

Mindenkinek jó tanulást! ☺

II. Motiváció

„Amit az emberi elme el tud képzelni, amiben hinni tud, azt meg is tudja valósítani”

(Napoleon Hill)

A számítástudomány és algoritmuselmélet első, vagy inkább 0-dik, vagy méginkább -1-edik mérföldköve Cook (1971) klasszikus eredménye, amelyben igazolja, hogy a 3-kielégíthetőségi probléma NP-teljes. Ez semmiképp sem jelenti azt, hogy itt bármiféle igényeket elégítenénk ki. Ez köznapin nyelven azt jelenti, hogy minden olyan igen-nem kérdés, amelyre adott válasz egy számítógéppel gyorsan ellenőrizhető, visszavezethető egy logikai 0-1 feladatra. Ez forradalmasította az algoritmus elméletet és előtérbe helyezte nemcsak mint tudományt, hanem mint feladatmegoldási technikát is a gráfelméletet és a kombinatorikát. Az első tételt hamarosan egy kötetrevaló követte (Garey et al., 1979). És míg ma egy ilyesfajta levezetés egyetemi zh-feladat, annak idején önálló, új, publikálható matematikai eredménynek számított. A gráfokat először Lovász László tanította Szegeden, a 70-es évek végén, 80-as évek elején. Itt, természetesen messze meghaladná a dolgozat terjedelmét Lovász professzor úr összes idevágó témájú munkájának idézése, de azt érdemes megjegyezni, hogy első három könyve mind számítástudományi témájú volt (Lovász, 1986, 1988, 1990). Innen a tudomány magyarországi elterjedése gyorsabb volt, mint a pestisé, vagy a Tisza áradása. Talán még Chuck Norrisnál is. 1981-ben már oktattak gráfokat Pesten is, 1983-ban már számítástudományi szempontból oktatta azokat Simonovits Miklós, 1984-ben bevitték a tanárszakos oktatásba és a 90-es évek elején megalakult az ELTE Számítógéptudományi Tanszéke. A műszaki oktatás sem kerülhette el a fejlődés (?) ezen irányát. Az Andrásfai Béla neve által fémjelzett villamosmérnöki diszkrét matematika tárgy (Andrásfai, 1968, 1971, 1973, 1977, 1983, 1984, 1991, 1997) 1994-ben hirtelen átcsapott (és azóta leülepedett) a Katona-Recski-Szabó trió által jegyzett, többtucat utánnyomást megért legendás klasszikusba. Maga a tantárgy is behatolt a természettudományi-műszaki-informatikai oktatás minden szintjére. Szegeden Hajnal Péter (2003), Lovász egyik első tanítványa vitte tovább a stafétát. A szegedi Info-kar megalakulása megszülte a tárgy Ésik Zoltán-féle felépítését is.

Sokan kérdezik, hogy mi értelme van ennyire mélyen belemenni különböző tantárgyakba, olyanokba, amit talán sosem fog használni az ember. Különösképpen szokott ez a kérdés

felmerülni matematika témájú tantárgyak tanulása közben. Erre a következőket szoktuk mondani:

Egyrészt, sosem tudhatja az ember, hogy mit hoz a jövő, hol lesz még szükség ezekre az ismeretekre. Másrészt az is könnyen előfordulhat, hogy valamilyen formában ebből fog megélni valamelyikünk.

Ennél véleményem szerint sokkal fontosabb, hogy valamilyen szinten megtanít küzdeni, ez jelenleg főleg a felsőfokú matematika tanulásunkra vonatkozik. Ha belegondolunk, hogy milyen kihívást jelent ilyen nehézségű témák „szakértőjévé” válni, egyből megértjük, hogy miről is van szó. Szerintünk hatalmas lelki erőt igényel, hogy valaki leüljön és a sok új ismeretet magáévé tegye, főleg akkor, ha az iskolai előadáson még semmit sem értett az anyagból. Tudni kell koncentrálni a könyvekre, még akkor is, ha a bennük szereplő információk elsőre felfoghatatlannak és rémisztőnek tűnnek. Az, hogy folytassuk a matematikai ismeretek elsajátítását, anélkül, hogy lemondóan becsuknánk a könyvet és feladnánk, egy dolgot jelent: „küzdelmet”. A harcot pedig életünk során gyakran fel kell vennünk.

Talán kézzel foghatóbbá válik, ha elmesélünk egy történetet a saját életünkből. Volt egy barátunk, nevezzük Janinak. Jani egy olyan középiskolából jött, ahol egyébként sem volt erős a matematika oktatás, és ő sem figyelt soha órán, végig kettesre teljesített. Amikor Jani bekerült a felsőoktatásba, lett egy Analízis nevű tantárgya. Mondanom sem kell, szenvedett vele. A csattanó az, hogy sikeres vizsgát tett félévkor, pedig ő szinte onnan kezdte a felkészülést, hogy: „Melyik is a számláló, és melyik a nevező?”. Nem viccelünk, szem- és fültanúk voltunk. Gondoljuk végig, hogy milyen kitartás és akaraterő kell ahhoz, hogy Jani szinte nulláról átmenjen Analízisből.

Ne feledjük, ne adjuk fel soha, bármilyen nehéz is a tananyag!

„A legfontosabb, amit a győzelemből megtanulhatunk, hogy képesek vagyunk rá.”

(Dave Weinbaum)

"A előbbre jutás nagy titka az, hogy el kell indulnod."

(Agatha Christie).

Induljunk el ...

III. Becslések

Elöljáróban gyorsan egy kicsit a közepéből, aztán részletesen megmagyarázva:

- **O :** Vagyis Nagy ordó.

Ha x elég nagy(nagyobb egy n_0 küszöbszámnál) és $|f(x)| \leq c \cdot |g(x)|$, ahol „ c ” lehet akármilyen nagy, de konstans kell legyen, akkor azt így jelöljük, és nagy ordónak mondjuk:

$$f(x) = O(g(x))$$

Ez az úgynevezett "felülről becslés".

- **Ω :** Vagyis Omega.

Az "alulról becslés" pedig így van: ha x tart végtelenbe és $|f(x)| \geq c \cdot |g(x)|$, akkor:

$$f(x) = \Omega(g(x))$$

- **Θ :** Vagyis Teta. Mondhatjuk úgy is, hogy a **Θ** a **Ω** és az **O** „párja”, függvények és számsorozatok nagyságrendjét jellemzi.

$$\boxed{\text{Fontos tudnunk, hogy } \Theta(g(n)) = \Omega(g(n)) \cap O(g(n))}$$

Lássuk tehát még egyszer, kicsit pontosabban az összefüggéseket.

Legyen f és g két, természetes számokon értelmezett valós függvény.

- I. Az $f = O(g)$ (f egyenlő nagy ordó g) jelölés tehát azt jelenti, hogy olyan $c > 0$ (pozitív, valós konstans) és olyan $n_0 > 0$ (pozitív küszöbszám), hogy minden $n \geq n_0$ esetén $|f(n)| \leq c \cdot |g(n)|$ (Minden olyan esetben, ahol az n szám nagyobb egyenlő, mint az n_0 küszöbszám)
- II. Az $f = \Omega(g)$ (f egyenlő omega g) azt jelenti, hogy van olyan $c > 0$ (pozitív valós konstans) és $n_0 > 0$ (pozitív küszöbszám), hogy minden $n \geq n_0$ esetén $|f(n)| \geq c \cdot |g(n)|$ teljesül.
- III. Az $f = \Theta(g)$ (f egyenlő teta g) jelölés pedig azt jelenti, hogy $f = O(g)$ és $f = \Omega(g)$ egyaránt teljesül. Vagyis **$\Theta(g(n)) = \Omega(g(n)) \cap O(g(n))$** !

Megjegyezhetjük, hogy $f = \Theta(g)$ pontosan akkor teljesül, ha van olyan $c_1, c_2 > 0$ valós konstans és $n_0 > 0$ küszöbszám, hogy minden $n \geq n_0$ esetén $c_1 \cdot |g(n)| \geq |f(n)| \geq c_2 \cdot |g(n)|$ teljesül.

Példák:

♣ Legyen $f(n) = 6n + 2000$ és $g(n) = n$.

- Ekkor $f = O(g)$, mert például $c = 7$ konstans esetén, és $n_0 = 2000$ választással $|f(n)| = f(n) = 6n + 2000 \leq 7n = c \cdot n$; ha $n \geq n_0 = 2000$.

Ellenőrzésként keressünk egy n értéket, ami a feltétel szerint nagyobb/egyenlő mint az n_0 küszöbszám. Legyen ez a szám a 3000. Behelyettesítve az egyenlőtlenséget azt kapjuk, hogy $20000 \leq 21000$, ami helyes.

Természetesen más jó választás is van a c konstansra. Az a lényeg, hogy vegyük észre, egy ilyen példánál olyan c számot választunk, ami nagyobb mint az n előtt álló konstans szorzó (ez azért is jó, mert $g(n)=n$; n az első hatványon van). Tehát példánkban egy olyan c számot kell választani, ami nagyobb 6-nál. Minden $c > 6$ értékhez találhatunk megfelelő n_0 értéket.

- $f = \Omega(g)$ is teljesül. Például válasszunk a következő képen: $c = 6$ és $n_0 = 1$. Ilyenkor $|f(n)| = f(n) = 6n + 2000 \geq 6n = c \cdot n$; ha $n \geq n_0$. Szintén belátható egy egyszerű behelyettesítéssel, hogy az egyenlőtlenség helyes.
- $f = \Theta(g)$ igaz, mivel $f = O(g)$ és $f = \Omega(g)$.

♣ Legyen $f(n) = 6n + 2000$ és $g(n) = n^2$

- Válasszunk úgy, hogy $c = 1$, a küszöbszám pedig maradhat $n_0 = 2000$. Ekkor $f = O(g)$, mivel $|f(n)| = f(n) = 6n + 2000 \leq 7n \leq n^2$ igaz, ha $n \geq n_0 = 2000$.

Tehát, választanunk kell egy n -t, ami nagyobb vagy egyenlő mint az n_0 küszöbszám, tehát mint 2000. Legyen ez a számunk az ellenőrzés kedvéért a 2500. Így n helyére beírva a 2500-as értéket azt kapjuk, hogy $17000 \leq 17500 \leq 6250000$, ami nyilván igaz. Az egyenlőtlenség helyességét egyszerűsítéssel láthatjuk be. (n -el osztunk)

- $f \neq \Omega(g)$, mert $|f(n)| = f(n) = 6n + 2000 \leq 7n$, és ezért ha $|f(n)| \geq c \cdot n^2$ teljesül, akkor ebből kifolyólag $7n \geq c \cdot n^2$ is igaz kell legyen. Ez pedig, akárhogy is választjuk a $c > 0$ számot, nem teljesül. Szintén könnyű megérteni, ha behelyettesítünk (és leosztunk n -el).

♣ Legyen $f(n) = 4n^2 - 90n + 6$

- Ebben az esetben $f(n) = O(n^2)$. Vegyük észre, hogy ha n megfelelően nagy, akkor $f(n) > 0$, így feltehetjük, hogy $|f(n)| = f(n)$. Ez biztos igaz, ha $4n^2 \geq 90n$, tehát mondjuk $n \geq 23$. Tehát csak az $n \geq 23$ helyeket nézzük meg. Ekkor $|f(n)| = f(n) = 4n^2 - 90n + 6 \leq 4n^2$, mivel $n \geq 23$. Azaz a $c = 4$ és az $n_0 = 23$ jó választás.
- $f(n) = O(n^3)$ is helyes, mert az előzőhöz hasonlóan itt is csak az $n \geq 23$ esettel foglalkozunk, amikor $|f(n)| = f(n) = 4n^2 - 90n + 6 \leq 4n^2 \leq n^3$, tehát a $c = 1$ és az $n_0 = 23$ jó választás. (23-at behelyettesítve $46 \leq 2116 \leq 12167$). Egyszerűsítsünk n -el, így adódik az igazunk.
- $f(n) \neq O(n)$, mert, ismét csak az $n \geq 23$ esetet tekintve láthatjuk, hogy $|f(n)| = f(n) = 4n^2 - 90n + 6 \geq 3n^2$, feltéve, hogy $n \geq 90$ is teljesül. Mivel $3n^2 > c \cdot n$ minden c -re, ha $n > c/3$, azt kapjuk, hogy $f(n) \geq c \cdot n$, ha n elég nagy.
- $f(n) = \Omega(n^2)$, hiszen az előbb láthattuk, hogy $f(n) \geq 3n^2$, ha $n \geq 90$.
- És természetesen $f(n) = \Theta(n^2)$, mert $f(n) = \Omega(n^2)$ és $f(n) = O(n^2)$ is teljesül.
- ♣ $2^{3n} \neq O(2^n)$, mivel $2^{3n} = 2^n \cdot 2^n \cdot 2^n$, tehát nagy n -ekre semmilyen c esetén sem teljesül, hogy $2^{3n} \leq c \cdot 2^n$

Most közelítsük meg egy nagy kicsit máshogyan ezt a dolgot, mondjuk úgy, hogy érdekesség képpen. A feladat a következő:

Igazolja, hogy $3n^3 + 2n^2 + n + 1 = \Theta(n^3)$!

Most bizonyítsuk be ezt a következő módon:

Majorálás:

$$f(n) = 3n^3 + 2n^2 + n + 1 \leq 3n^3 + 2n^3 + n^3 + n^3 (= 7n^3); \text{ (ha } n \geq 1)$$

$$\text{Vagyis } f(n) = O(n^3)$$

Nézzük, hogy mit is csináltunk, nagyon a lényegre törően. Gyártottunk egy függvényt, ami biztos hogy **nagyobb**, de legalább egyenlő az eredeti függvényünkkel. (ha az n -re tett kikötés teljesül) Ezt úgy csináltuk, hogy az f függvényben minden n -es, és minden csak konstans tartalmazó tagot kicseréltünk az eredeti függvényben található legnagyobb hatványon lévő n -re, úgy, hogy az előtte álló szorzószámokat meghagytuk. Tehát a feladatban mindenhova n^3 került, de a 2-es, (és az 1-es) szorzószám megmaradt. Aztán kikötést kell tenni

az n -re, hogy ez igaz legyen, és ez pedig akkor lesz igaz, ha $n \geq 1$. Aztán kimondhatjuk, hogy $f(n) = O(n^3)$.

Minorálás:

$$f(n) = 3n^3 + 2n^2 + n + 1 \geq 3n^3 ; (ha\ n \geq 1)$$

$$Vagyis\ f(n) = \Omega(n^3)$$

Tulajdonképpen nem nehéz kitalálni, hogy mi is történt most. Most a nagyobb egyenlő jel másik oldalára egy olyan függvény került, ami biztos, hogy **kisebb, vagy egyenlő** mint az eredeti függvényünk (ha az n -re tett kikötés teljesül). Erre a célra alkalmas az f függvény első, legnagyobb hatványon lévő n -es tagja, konstans szorzóval együtt, tehát a $3n^3$. Szintén teszünk kikötést az n -re. Majd kimondhatjuk, hogy $f(n) = \Omega(n^3)$.

A könnyebb átláthatóság kedvéért írjuk egy sorba az egészet, tehát:

$$3n^3 \leq 3n^3 + 2n^2 + n + 1 \leq 3n^3 + 2n^3 + n^3 + n^3 (ha\ n \geq 1)$$

Ha ez megvan, már nem sok dolgunk van, mint szépen leírni a következőt:

$$Mivel\ \Theta(n^3) = \Omega(n^3) \cap O(n^3),\ ezért\ igaz,\ hogy\ 3n^3 + 2n^2 + n + 1 = \Theta(n^3).$$

[Ugrás a feladatokhoz!](#)

IV. Igazságtáblák

Az úgynevezett állításokat, kijelentéseket A-val, B-val, C-vel stb. jelöljük.

Logikai értéküket tekintve az állítások lehetnek igazak, vagy hamisak. Ha az állítás

♣ igaz, azt jelölheti: 1 , i , $\uparrow$

♣ hamis, azt jelölheti: 0, h , $\downarrow$

Fontos tudnunk, hogy minden állításnak kell hogy legyen egy meghatározott logikai értéke (igaz vagy hamis), valamint két állítás egyenlő, ha logikai értékük egyenlő.

Ismerünk úgynevezett logikai műveleteket, ezek a következők:

♣ „És” művelet, konjunkció: $A \wedge B$; $A \cdot B$

♣ „Vagy” művelet, diszjunkció: $A \vee B$; $A + B$

♣ Tagadás, negáció: $\overline{A}$

♣ Implikáció: $A \Rightarrow B$

♣ Ekvivalencia: $A \Leftrightarrow B$

Egyezünk meg abban, hogy az igaz ítélet jele mindig 1 lesz, a hamisé pedig mindig 0. Ennek tudatában nézzük meg igazságtáblával, hogy milyen értékeket vesznek fel a logikai műveleteink.

A	1	1	0	0
B	1	0	1	0
$A \wedge B$	1	0	0	0
$A \vee B$	1	1	1	0
$\overline{A}$	0	0	1	1
$A \Rightarrow B$	1	0	1	1
$A \Leftrightarrow B$	1	0	0	1

(4.1-es táblázat)

Most nézzük sorban, hogy miért így alakultak az „ítéletek”.

- $A \wedge B$: Akkor és csak akkor *igaz*, ha A igaz, és B is igaz
- $A \vee B$: Akkor és csak akkor **hamis**, ha A hamis, és B is hamis
- $\bar{A}$: Mindig az A ellentéte
- $A \Rightarrow B$: Akkor és csak akkor **hamis**, ha A igaz, viszont B hamis
- $A \Leftrightarrow B$: Akkor és csak akkor *igaz*, ha $A = B$

Fontos dolog következik, ezt jegyezzük meg jól:

$$A \Rightarrow B = \bar{A} \vee B$$

$$A \Leftrightarrow B = (A \Rightarrow B) \wedge (B \Rightarrow A) = (A \wedge B) \vee (\bar{A} \wedge \bar{B})$$

Most igazoljuk ezeket az egyenlőségeket igazságtáblával!

A	1	1	0	0
B	1	0	1	0
$\bar{A}$	0	0	1	1
$\bar{A} \vee B$	1	0	1	1
$A \Rightarrow B$	1	0	1	1

(4.2-es táblázat)

A	1	1	0	0
B	1	0	1	0
$\bar{A}$	0	0	1	1
$\bar{B}$	0	1	0	1
$\bar{A} \wedge \bar{B}$	0	0	0	1
$A \Rightarrow B$	1	0	1	1
$B \Rightarrow A$	1	1	0	1
$A \wedge B$	1	0	0	0
$(A \Rightarrow B) \wedge (B \Rightarrow A)$	1	0	0	1
$(A \wedge B) \vee (\bar{A} \wedge \bar{B})$	1	0	0	1
$A \Leftrightarrow B$	1	0	0	1

(4.3-as táblázat)

Ezen igazságtáblák megoldásait nem magyaráznám most el, nem egy nagy művészet, remélem így is érthető.

Diszjunktív normálforma

Tétel: Minden logikai művelet megadható a konjunkció, diszjunkció, negáció alkalmazásával.

Itt átvennék egy konkrét példát a 2011-es Hornung tanár úr féle Számítástudomány pdf-ből.

A	1	1	1	1	0	0	0	0
B	1	1	0	0	1	1	0	0
C	1	0	1	0	1	0	1	0
f(A,B,C)	0	1	0	0	0	1	0	1

(4.4-es táblázat)

$$f(A,B,C) = (A \wedge B \wedge \bar{C}) \vee (\bar{A} \wedge B \wedge \bar{C}) \vee (\bar{A} \wedge \bar{B} \wedge \bar{C})$$

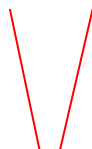
Definíció: Negált vagy negálatlan változók konjukciójának diszjunkcióját *diszjunktív normál formának* nevezzük.

Érdemes kétszer-háromszor elolvasni ezt a definíciót ☺

Természetesen eredeti formájában kell tudni, de most nézzük meg, hogy mit is mond, mi is az a diszjunktív normál forma? Tehát negált vagy negálatlan változók, azaz $A, B, C, \dots$ vagy $\bar{A}, \bar{B}, \bar{C}, \dots$ konjukciójának diszjunkcióját, azaz ezen változók olyan formáját, ahol „vagy” művelettel összekapcsolt „és” műveletek vannak.

$$f(A, B, C) = (A \wedge B \wedge \bar{C}) \vee (\bar{A} \wedge B \wedge \bar{C}) \vee (\bar{A} \wedge \bar{B} \wedge \bar{C})$$

Mint látjuk, a negálatlan, és/vagy negált változók össze vannak kapcsolva az „és” művelettel.



A definíció következő része pedig itt látszik, látjuk hogy a zárójelben lévő kifejezések pedig a „vagy” művelettel vannak összekapcsolva. Így talán már érthetőbb, hogy mi is az a diszjunktív normál forma

V. Halmazelmélet, Rendezések, Relációk

Jó, ha tudjuk középiskolából, hogy a halmazokat a latin ábécé nagy betűivel jelöljük, tehát A,B,C...

Azt, hogy „valami” eleme, vagy nem eleme egy halmaznak, úgy jelöljük, hogy $x \in A$; $x \notin B$.

Továbbá tudjuk, hogy minden halmazról egyértelműen eldönthető, hogy mik az elemeik, valamint azt hogy két halmaz egyenlő, ha elemeik megegyeznek.

Halmazok megadhatunk, úgy hogy

- felsoroljuk az elemeit: $A = \{a,b, \dots\}$
- megadjuk a tulajdonságát: $A = \{x \in H \mid P(x)\}$ (= P igazsághalmaza)

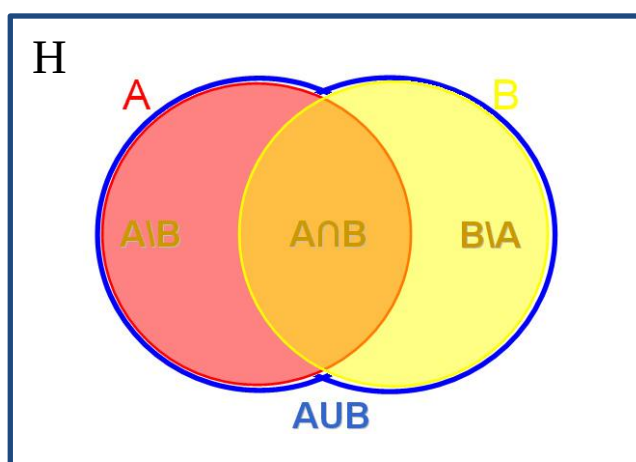
Az A halmaz akkor részhalmaza a B halmaznak, ha A minden eleme B-nek is eleme. Jelölés: $A \subseteq B$. Megjegyezhetjük, hogy $A = B$ akkor, ha $(A \subseteq B \text{ és } B \subseteq A)$.

Az A halmaz akkor **valódi** részhalmaza B halmaznak, ha az A halmaz *reszhalmaza* B-nek, és a két halmaz elemei nem egyeznek meg. Tehát $A \subset B \Leftrightarrow (A \subseteq B \wedge A \neq B)$.

A halmazokkal végezhetünk különböző műveletek, ezek a következők:

- Metszet, közös rész: $A \cap B = A \cdot B := \{x \mid x \in A \wedge x \in B\}$
- Egyesítés, unió: $A \cup B = A + B := \{x \mid x \in A \vee x \in B\}$
- Komplementer: $\bar{A} := \{x \in H \mid x \notin A\}$, H az alaphalmaz
- Különbség: $A \setminus B = A - B := A \cap \bar{B}$

Érdemes szemléltetni ezeket a műveleteket Venn-diagrammal:



(5.1-es ábra)

Akkor most vágjunk bele a relációkba!

Definíció: Az $A_1, \dots, A_n$ halmazok e sorrendben vett Descartes-féle szorzata, vagy röviden szorzata az

$$A_1 \times \dots \times A_n := \{(a_1, \dots, a_n) \mid a_i \in A_i, \forall i = 1, \dots, n\}$$

halmaz. Speciálisan, ha $A = A_1 = \dots = A_n$, akkor az $A^n := A_1 \times \dots \times A_n$ egy úgynevezett homogén Descartes-féle szorzat.

Lássuk ezt egy kicsit barátságosabb nyelven. A következő sor szerintem egy kis magyarázatra szorul:

$$A_1 \times \dots \times A_n := \{(a_1, \dots, a_n) \mid a_i \in A_i, \forall i = 1, \dots, n\}$$

Azt jelenti, hogy a halmazok úgynevezett Descartes-féle szorzata egy olyan halmaz, melynek a tagjai az $(a_1, a_2, a_3, \dots, a_n)$ -el jelölt elemek, ahol is a_i (i -edik halmazból származó tag) az A_i (i -edik halmaz) eleme, és az i -k 1-től mennek egyesével n -ig (n elméletben lehet akár 17563893 is).

Azt, hogy a halmazok elemei között valamilyen kapcsolat áll fenn, a reláció fogalma ragadja meg. Jelölje A egy házibuli fiúinak a halmazát, míg B a buliban lévő lányok halmaza legyen. Ekkor az $R \subseteq A \times B$ halmazzal megadhatjuk, hogy ki kivel fog „összebarátkozni”. De akár azt is mondhatjuk, hogy ha van két halmazunk, $A = \{\text{Sör, Bor, Vodka, Whiskey, \dots, Pezsgő}\}$ és $B = \{\text{Kóla, Narancslé, Málnaszörp, Fanta, \dots, Ásványvíz}\}$. Ha a két halmaz relációban áll, akkor meghatározzák, hogy milyen italokat lehet kikeverni. Tehát $A \times B$ halmaz megadja például a vodkanarancsot vagy a Whiskey-kólát, de ez maradjon köztünk. ☺

A reláció fogalma nemcsak két halmaz közti viszony megadására szolgál.

Definíció: Legyenek adottak az $A_1, \dots, A_n$ halmazok. Ekkor minden $R \subseteq A_1 \times \dots \times A_n$ halmaz meghatároz egy $A_1, \dots, A_n$ fölötti (n -áris) relációt.

„Az adatbázisok például n -áris relációknak tekinthetők. Legyen például A_1 az országok halmaza, A_2 a települések halmaza, $A_3 := \{\text{főváros, egyéb város, egyéb település}\}$ és $A_4 := \mathbb{N}$ a népesség száma ezer főben. Tartalmazza az R „adatbázis” (reláció) a világ településeinek adatait. Ekkor például $(\text{Magyarország, Budapest, főváros, 1740}) \in R$.

A Descartes-féle szorzat definíciójához hasonlóan egy $R \subseteq A^n$ relációt **homogénnek** nevezünk, különben pedig **inhomogén** relációról beszélünk. Az $R \subseteq A^n$ homogén relációt egyszerűen A feletti n -áris relációnak mondjuk.

Hangsúlyozzuk, hogy egy relációt nemcsak az elemek közötti viszonyok határoznak meg, hanem azok a halmazok is, amelyek fölött a reláció értelmezett.

A speciálisan két halmaz elemei közötti viszonyt megadó relációkat **bináris** relációknak nevezzük.” (Tasnádi Attila: Számítástudomány Gazdaságinformatikusoknak, Bologna – Tankönyvsorozat, Budapest 2006.)

Definíció: Az $R \subseteq A \times B$ egy A és B fölötti bináris reláció. Egy $a \in A$ elem és egy $b \in B$ elem egymással R relációban van, ha $(a,b) \in R$.

A definíció megmagyarázására most inkább nem térünk ki, majd a konkrét példánál tökéletesen érthető válik (bár reméljük már most is az ☺)

A kisebb-nagyobb reláció példa egy bináris relációra. Ennek mintájára sokszor $(a,b) \in R$ helyet használjuk az aRb írásmódot is.

Homogén bináris relációk

Sokak öröme (esetleg bánatára) mi csak a homogén bináris relációkkal foglalkozunk.

Először is nézzük meg a tulajdonságait. Tehát az $R \subseteq A^2$ homogén bináris reláció

- $\Rightarrow$ **reflexív**, ha aRa teljesül minden $a \in A$ elemre;
- $\Rightarrow$ **szimmetrikus**, ha minden $a,b \in A$ elemekre abból, hogy aRb teljesül, bRa is következik;
- $\Rightarrow$ **antiszimmetrikus**, ha minden $a,b \in A$ elemekre az aRb és bRa egyidejű teljesülése esetén $a = b$;
- $\Rightarrow$ **transzítív**, ha minden $a,b,c \in A$ elemekre az aRb és bRc egyidejű teljesülése esetén aRc következik;
- $\Rightarrow$ **teljes**, ha minden $a,b \in A$ elemekre aRb vagy bRa teljesül.

Most egyszer nézzük végig a témához kapcsolódó összes fontosabb állítást, definíciót, aztán nézzünk rá példákat.

Definíció: Az R homogén bináris relációt **ekvivalenciareláció**nak nevezzük, ha *reflexív*, *szimmetrikus* és *transzitiv*.

Definíció: Az R homogén bináris relációt **parciális rendezés**nek nevezzük, ha *reflexív*, *antiszimmetrikus* és *transzitiv*.

Definíció: Az R parciális rendezés egy **teljes rendezés** vagy más néven **lineáris rendezés**, ha *teljes* is.

A teljes rendezések eleget tesznek az úgynevezett trichotómia tulajdonságnak.

Definíció: Az A fölötti R homogén bináris relációt **trichotómnak** nevezzük, ha tetszőleges $a, b \in A$ esetén

1. $a = b$
2. $a \neq b$ és aRb
3. vagy $a \neq b$ és bRa

tulajdonságok pontosan egyike teljesül.

Állítás: Az R parciális rendezés pontosan akkor teljes rendezés, ha trichotóm.

Definíció: Az R homogén bináris relációt **preferenciareláció**nak nevezzük, ha *reflexív*, *transzitiv* és *teljes*.

Megfigyelhetjük, hogy a teljesség maga után vonja a reflexivitást.

Mielőtt elkezdenéd nézni a feladatokat Kedves Olvasó, javaslom hogy előbb az elméleti részeket tanuld meg, mert csak úgy lesz értelme ☺

Példa feladat:

Legyen $A = \{1,2,3,4,5,6\}$! Értelmezzük az $R \subseteq A \times A$ binér relációt úgy, hogy aRb akkor és csak akkor áll fenn $a, b \in A$ esetén, ha

- a) b osztható a -val
- b) $(a-b)$ osztható 2-vel.

Melyik igaz, illetve hamis a következő állítások közül:

- R ekvivalencia reláció A -n,
- R parciális rendezés A -n,
- R teljes rendezés A -n!

Válaszát indokolja!

Ábrázolja a $G=(A, R)$ irányított gráfot!

Megoldás:

a, Először is írjuk fel az összes olyan párosítást a számokban, amikre igaz, hogy b osztható a -val. Tehát olyan számpárok kellenek, ahol a második helyen álló számban megvan az első helyen álló szám maradék nélkül. Ilyen számpárok a következők: $(1,1);(1,2);(1,3);(1,4);(1,5);(1,6);(2,2);(2,4);(2,6);(3,3);(3,6);(4,4);(5,5);(6,6)$. Tehát a jobb oldali szám (b) osztható a bal oldali számmal (a).

Most el kell döntenünk, hogy melyik igaz és melyik hamis az alábbi állítások közül:

R ekvivalencia reláció-e? Az ekvivalencia relációról tudjuk, hogy reflexív, szimmetrikus és tranzitív. A reflexivitás teljesül, hiszen minden szám osztója saját magának, ugye aRa teljesül minden $a \in A$ elemre.

Nem lehet szimmetrikus, mivel a szimmetria azt mondja, hogy minden $a,b \in A$ elemekre abból, hogy aRb teljesül, bRa is következik. Itt viszont például abból, hogy a 2 osztója a 4-nek, nem következik, hogy a 4 osztója a 2-nek. A felsorolásunkban is csak a $(2,4)$ szerepel, a $(4,2)$ nem.

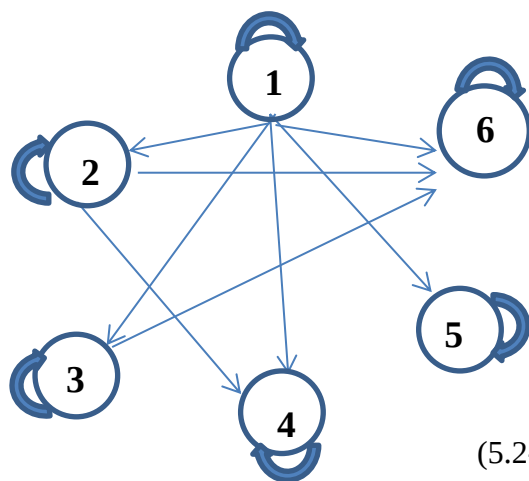
Elárulom, hogy a tranzitivitás teljesül. Vegyünk a halmazunkból 3 számot, legyenek ezek az 1,2,4. A reláció tranzitív, ha minden $a,b,c \in A$ elemekre az aRb és bRc egyidejű teljesülése esetén aRc következik. Tehát ha azt mondjuk, hogy az 1 osztója a 2-nek, és a 2 osztója a 4-nek, abból következik, hogy a 1 osztója a 4-nek. $(1,2)$ és $(2,4)$ szerepel a felsorolásban, tehát ennél a példánál az $(1,4)$ párosításnak is szerepelnie kell. Tehát ez nem ekvivalencia reláció.

R parciális rendezés-e? A reláció akkor parciális rendezés, ha reflexív, antiszimmetrikus és tranzitív. Az előbb láttuk, hogy a reflexivitás és a tranzitivitás teljesül, kérdésnek marad az antiszimmetria. A reláció akkor antiszimmetrikus ha minden $a,b \in A$ elemekre az aRb és bRa egyidejű teljesülése esetén $a = b$. Tehát mondjuk a ha a jobb oldali számunk (b) osztható a bal

oldali számmal (a), és ugyanez fordítva is igaz, hogy a is osztható b -vel, csak akkor teljesül, ha a két szám egyenlő. Vegyük észre a felsorolásunk segítségével, hogy ez tényleg csak ilyen esetben igaz, a többiben nem. Például $(2,2);(4,4)$. Vagyis a reláció parciális rendezés.

R teljes rendezés-e? Akkor teljes a parciális rendezés, ha minden $a, b \in A$ elemekre aRb vagy bRa teljesül, vagyis esetünkben akkor, ha bármelyik két szám amit kinézünk a halmazból osztója egymásnak, vagy úgy hogy az a lesz bal oldalt és a b jobb oldalt, vagy fordítva. Tehát például ha nézzük a 4-est és a 6-ost. Akkor lenne teljes a rendezés, ha a 4 osztója lenne a 6-nak, vagy a 6 lenne osztója a 4-nek. Ez nem igaz, a felsorolásunkban sincsen se $(4,6)$, se $(6,4)$, vagyis a rendezés nem teljes.

Most ábrázolni kell a $G=(A,R)$ irányított gráfot. A gráfelmélet igazából a következő témakör, de ez a feladat nem nagy művészet. Annyi a dolgunk, hogy csúcsoknak felvesszük a **halmazunk számait** ($G=(A,R)$), a csúcsok között az **összekötő vonalak** pedig a „kapcsolatok”, a relációk ($G=(A,R)$). Így néz ki:



Mivel a gráf irányított, ezért a csúcsokat nyíllal, és nem sima vonallal kötjük össze. A hurokél jelenti, hogy a szám osztója saját magának.

(5.2-es ábra)

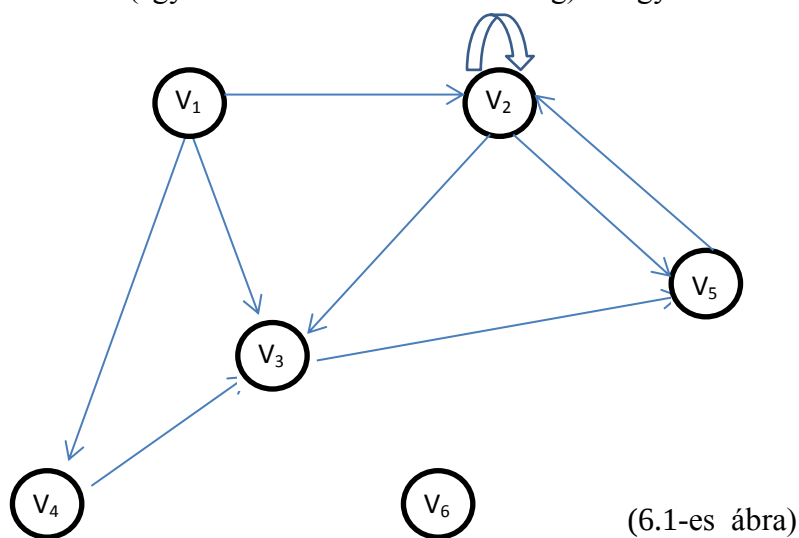
b, Lásd a mellékletben ☺

[Ugrás a feladatokhoz!](#)

VI. Gráfelmélet és Gráfalgoritmusok

Mielőtt belevágunk a témába, tudnunk kell, hogy ami ebben a könyvben van elmélet címszó alatt, az kevés, de nincs értelme ide leírni minden tételt, definíciót, ami kérdés lehet, csak a lényegesekkel foglalkozunk. Főleg azokkal, amik közvetlenül kellenek a feladat megoldásokhoz.

Reméljük mindenki tudja, hogy mi fán terem egy gráf, hogyan is néz ki. A gráfokat a csúcsaik, és az éleik segítségével adjuk meg, $G=(V, E)$ formában. V a csúcsok halmaza, E pedig az élek halmaza (egy élt két csúccsal adunk meg). A gyakorlatban ez a következő képpen néz ki:



Ez a $G=(V,E)$ irányított gráf, ahol

$V = \{v_1, v_2, \dots, v_6\} \rightarrow$ Ez a csúcsok halmaza

$E = \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_2, v_2), (v_2, v_3), (v_2, v_5), (v_3, v_5), (v_4, v_3), (v_5, v_2)\} \subseteq V \times V = V^2 \rightarrow$ Élek halmaza, egy él a kezdő és a végponttal van megadva.

A gráfban egy hurok él van a v_2 csúcsnál, többszörös él nincsen benne.

A tananyagban innentől kezdve hihetetlen mennyiségű elméleti tudnivaló, definíció és tétel a „jutalom”, ezeket természetesen nem árt a tankönyvből vagy az órai anyagból megtanulni, egy-két nagyon lényeges dolog fog csak bekerülni ebbe a könyvbe. Szóval lássuk sorban a definíciókat.

Gráfokkal kapcsolatos alapfogalmak

Definíció:

- Egy $G = (V, E)$ gráf két $u, v \in V$ csúcsát szomszédosnak mondjuk, ha $uv \in E$. Vagyis akkor szomszédos két csúcs, ha ugyan azon az élen vannak rajta.
- Két $e, f \in E$ élt szomszédosnak nevezzük, ha van közös csúcsuk.
- A $v \in V$ csúcs illeszkedik az $e \in E$ éltre, ha létezik olyan $w \in V$ csúcs, hogy $e = vw$. Tehát a v csúcs akkor van rajta az e élen, ha az élnak a v csúcs a kezdő vagy a végpontja.
- Az $e \in E$ él illeszkedik a $v \in V$ csúcsra, ha létezik olyan $w \in V$ csúcs, hogy $e = vw$. Vagyis az e élnak a v csúcs kezdő vagy végpontja, ha a másik pont a w csúcs, ami meghatározza az élt (az élnak kezdő vagy végpontja a w csúcs).
- A $v \in V$ csúcs egy izolált csúcs, ha nincsen vele szomszédos másik csúcs. (Az ábrán a v_6 ilyen)

Definíció: Egy $G = (V, E)$ gráf egy $v \in V$ csúcsának fokszáma a v csúcsra illeszkedő élek száma, ahol a $vv \in E$ hurokért duplán számoljuk. A $v \in V$ csúcs fokszámát $\deg(v)$ -vel jelöljük.

- ✓ G maximális fokszáma:

$$\Delta(G) := \max\{\deg(v) \mid v \in V\}$$

- ✓ G minimális fokszáma:

$$\delta(G) := \min\{\deg(v) \mid v \in V\}$$

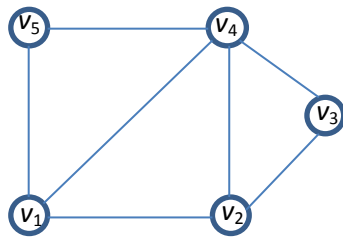
A $G = (V, E)$ gráfra igaz, hogy $\sum_{v \in V} \deg(v) = 2|E|$, ahol $|E|$ a G gráf éleinek számát jelöli.

És így tovább, még rengeteg elmélet... De az elméletből most elég lesz ennyi. ☺

A gráfalgoritmusok közül a következőkről fog szó esni:

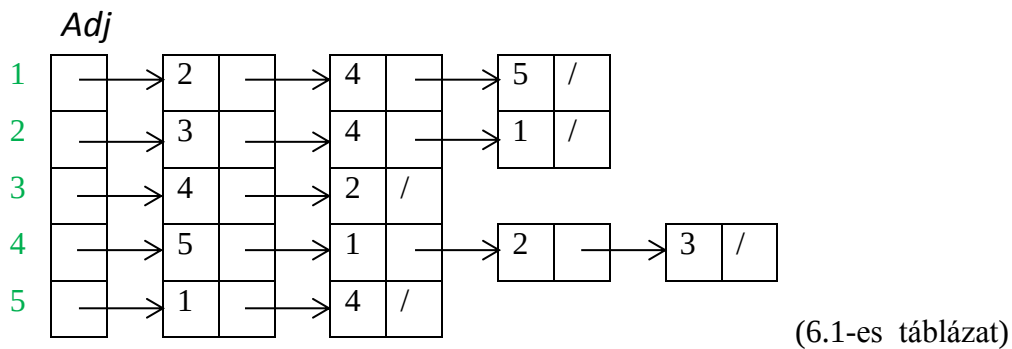
- ✓ Szélességi bejárás
- ✓ Mélységi bejárás
- ✓ Kruskal algoritmusa
- ✓ Dijkstra algoritmusa
- ✓ A Gale-Shapley-algortimus
- ✓ Hálózati folyamatok

Mielőtt ezekről beszelnénk, nézzük meg a szomszédsági listát és a szomszédsági mátrixot.



(6.2-es ábra)

Szomszédsági listával a következő módon adjuk meg ezt a gráfot:



A gráf nem irányított, vagyis nem rajzolunk nyilat a vonalak végére. Ezért így néz ki a lista, megmutatja, hogy **melyik csúcsból** melyik csúcsba megy el.

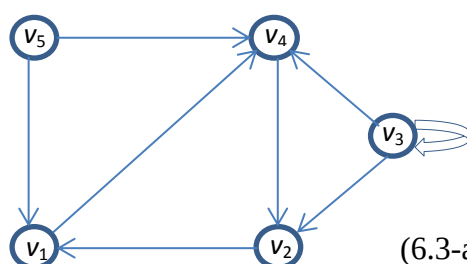
Ugyanennek a gráfnak a szomszédsági mátrixa:

A	1	2	3	4	5
1	0	1	0	1	1
2	1	0	1	1	0
3	0	1	0	1	0
4	1	1	1	0	1
5	1	0	0	1	0

(6.2-es táblázat)

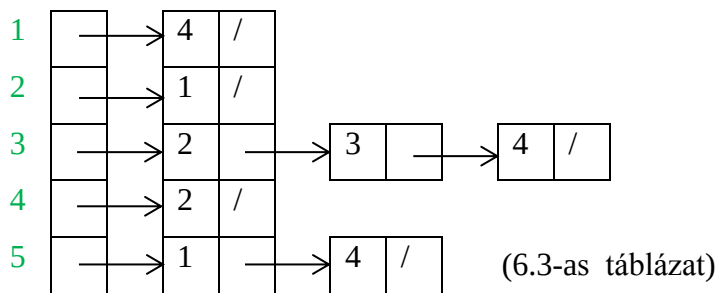
A félkövér számok a csúcsokat jelentik. Ha a két szám metszetében 1-es van, akkor az a két csúcs össze van kötve, ha pedig 0, akkor nincs.

Ha a gráfunk irányított, akkor:



(6.3-as ábra)

Az irányított gráf szomszédsági listája:



Ha irányított a gráf, akkor a nyilak miatt pontosan tudjuk, hogy a csúcsok között melyik irányba lehet „menni” az élen. Ez nemsokára fontos lesz.

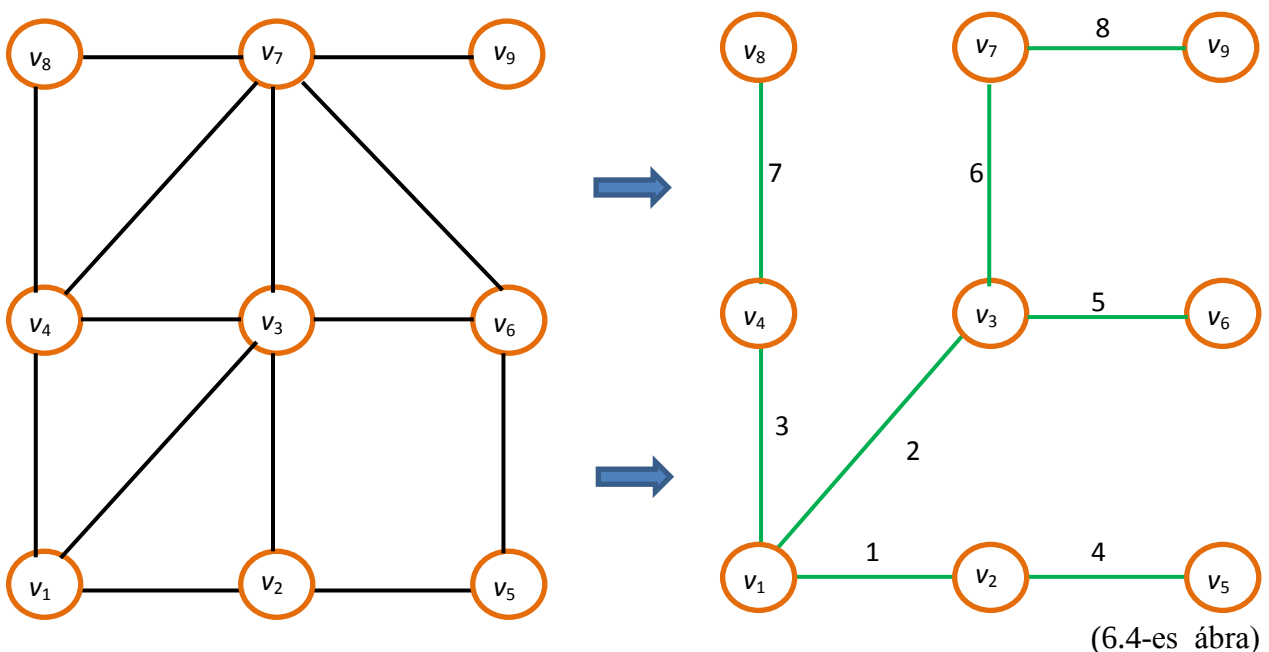
Az irányított gráf szomszédsági mátrixa:

A	1	2	3	4	5
1	0	0	0	1	0
2	1	0	0	0	0
3	0	1	1	1	0
4	0	1	0	0	0
5	1	0	0	1	0

(6.4-es táblázat)

Szélességi bejárás

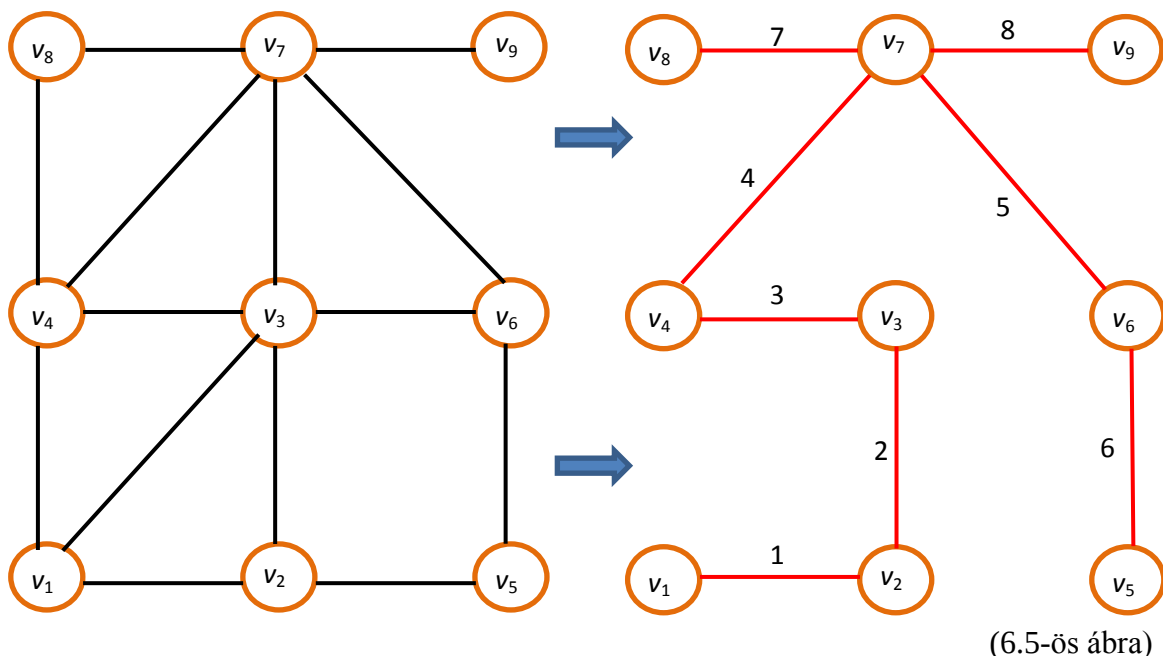
A gráfok bejárásának egyik módja a szélességi bejárás. A bejárásnak az a lényege, hogy egy kezdő csúcsból (ez a v_1 lesz nálunk) el tudjunk jutni az összes többi csúcsba valamilyen módon. Szélességénél az a lényeg, hogy egy adott csúcsból először elmegyünk az összes szomszédos csúcsba, majd ha ezek megvannak, akkor visszalépünk egyik csúcsba (maradjunk abban, hogy abba a csúcsba lépünk vissza, amibe először elmentünk az „induló” csúcsból) és abból is megcsináljuk ugyan ezt, amíg nem érjük el az összes pontot. Tehát:



Először is, az éleken lévő szám azt jelenti, hogy hányadikként választottuk ki azt az élt. Az 1-es csúcsban kezdünk, majd elmegyünk az összes szomszédjába, először a kettesbe, aztán a hármasba, majd a négyesbe. Mivel az első csúcsnak nincs több szomszédja, ezért a másik pont szomszédjait kezdjük el vizsgálni. Először a második csúcsba jutottunk el, ezért az lesz a következő, aminek megnézzük a szomszédjait (pár sorral feljebb erre próbáltam utalni, hogy így fogunk haladni ilyen feladatnál). Negyedik lépésként eljutunk a kettesből az ötösbe, és mivel nincs több szomszédja a kettesnek, amiben még nem voltunk, új csúcsot keresünk. Ez a 3-as lesz, mivel az volt az amibe másodikként mentünk. Aztán, ha vezet út mindkét szomszédjába, jön a v_4 , aminek egy szomszédja van, oda húzunk egy élt, ez a hetedik lépésünk. Utolsó lépés, hogy utat csináljunk oda, ahol még nincs, tehát a v_9 -be, ez pedig csak egy módon lehet ebben a helyzetben. Remélem érthető ☺

Mélységi bejárás

Mélységi bejárásnál ha a kezdő csúcsból elmentünk egy szomszédos csúcsba, akkor onnan rögtön továbbmegyünk a következőbe, nem nézzük meg a kezdő csúcs összes szomszédját. Ha pedig már „nincs hova menni”, akkor visszalépünk addig a csúcsig, ahonnan még van lehetőségünk mozogni. Így néz ki:



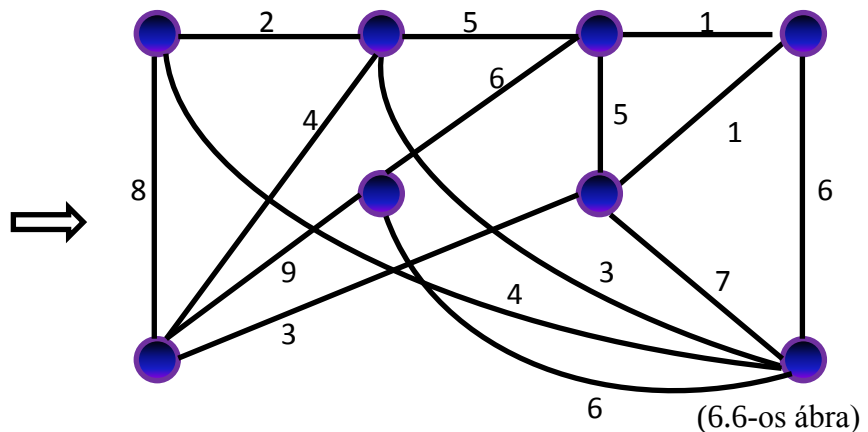
Szóval, az első csúcsba indulunk, aztán elmegyünk a kettesbe. Nem nézzük meg az első csúcs többi szomszédját, hanem azonnal tovább megyünk a kettesből a hármasba (ha lehetőség van rá, tartsuk a számok sorrendjét, ezért nem az ötösbe megyünk), onnan pedig 4-

es,7-es, és a 6-os felé megyünk, onnan pedig az 5-ösbe. Mivel nem tudunk innen továbblépni, ezért visszamegyünk egyet. A hatosból nem tudunk már értelmes lépést csinálni, ezért még egyet visszalépünk a hetesbe. Onnan pedig 8-as, majd 9-es.

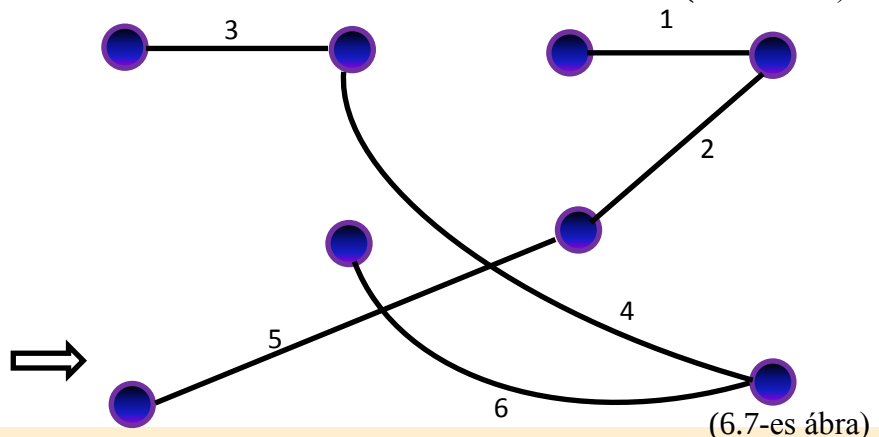
Kruskal algoritmus

Röviden elmondva itt arról lesz szó, hogy a gráfból csinálni kell egy úthálózatot, amin minden csúcsba el lehet jutni. Azonban itt már az éleknek van úgynevezett súlya, aminek az értéket ráírjuk az élre, szóval itt már az élen lévő szám nem a kiválasztás sorrendjét fogja jelölni. Kruskal algoritmusát a minimális feszítőfa meghatározására használjuk, tehát az a cél, hogy bármelyik csúcsból el lehessen jutni akármelyik másikba, a lehető legkisebb összsúly mellett. Van egy nagyon jó példa Tasnádi Attila: Számítástudomány gazdaságinformatikusoknak című könyvében, a 45. oldalon. Mi is ezt nézzük meg.

Az éleken szereplő szám tehát azt jelenti, hogy mekkora az él súlya. Ebből a gráfból kell csinálni egy úgynevezett minimális feszítőfát, tehát minden csúcsba vezessen út minden pontba, de ha összeadjuk a felhasznált éleken lévő számokat, akkor azok a lehető legkisebb értéket vegyék fel.



Először is, itt már az élen lévő szám azt mutatja, hogy hányadikként választottuk ki az élt. Nézzük végig, hogy miért ez a megoldás. Először, megkeressük azt az élt, aminek a legkisebb a súlya, ez az 1 súlyú él, ezt behúzzuk. Aztán megnézzük, hogy van-e még olyan, aminek egy a súlya. Mivel van ilyen, és olyan élbe megy, amit még nem tudunk elérni, ezért azt is berajzoljuk az új ábránkra. Most keresünk olyan éleket, amiknek 2 a súlya, ilyen 1 van, ezt is berajzoljuk, mivel szintén olyan csúccsal bővíti a fánkat, amibe eddig még nem juthattunk el.

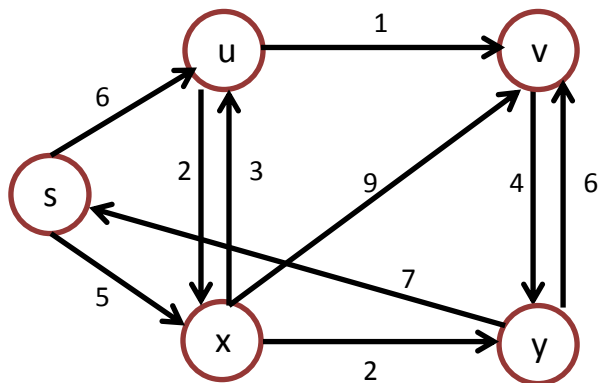


Ezután jönnek a 3 súlyú élek. Rajzoljuk be mindkettőt, ugyan azért amiért az eddigieket is. Megyünk tovább, 4 súlyú élekkel rendelkezünk, arra is szükségünk van. Most jönnek az 5 súlyú élek, egy darab van. Látjuk, hogy azt fölösleges berajzolni, mivel olyan csúcsokat kötne össze, ahova már vezet út, tehát fölösleges lenne nekünk ez az él. Nézzük tovább, jön a 6-os. Három darab ilyen él van, de ebből elég egyet berajzolni, és kész a feladat, mivel minden csúcsba van út. Nem lehet kör a feszítőfában!

Kész van a minimális összsúlyú feszítőfa, összsúlya 20.

Dijkstra algoritmus

Ez az algoritmus arra való, hogy segítségével meghatározzuk bármelyik rögzített csúcsból az összes többibe vezető legrövidebb út hosszát. Ehhez ismernünk kell a szomszédos csúcsok távolságát. Nézzünk ehhez egy gráfot:



(6.8-as ábra)

Ez egy elég egyszerű példa, az a cél, hogy az s csúcsból el lehessen jutni az u,x,v,y csúcsba. Ezt úgy kell elérni, hogy ha összeadjuk a felhasznált éleken lévő számokat, akkor a lehető legkevesebb legyen az összeg. A megoldást táblázatos formában is meg kell adni, plusz be kell rajzolni az eredeti gráfra a felhasznált éleket. (Mi újat rajzolunk itt.)

Megoldása:

s	x	y	u	v
0,NIL	∞	∞	∞	∞
	5,s	∞	6,s	∞
		7,x	6,s	14,x
		7,x		7,u
				7,u

(6.5-ös táblázat)

3. Tehát, az első sort megbeszéltük, hogy az mindig így van. Következik a második sor, melyben megnézzük, hogy az s csúcsból hova tudunk eljutni. Eljuthatunk az x-be, ezért beírjuk, hogy „5,s”, mivel az út hossza 5 és az s pontból jöttünk. El tudunk még menni az s-ből az u-ba, az út hossza 6, beírjuk ezt is. Az y és a v csúcs nem szomszédos az s-el, ezért oda végtelen jel kerül. Most az a kérdés, hogy melyik csúcsból vizsgálódjunk tovább? Mindig azt kell választani, amelyikbe rövidebb úton érkezhünk, Tehát most az 5 és a 6 közül kell választani.

1. Az első sorban felvesszük az összes csúcsot, egy csúcs = egy oszlop

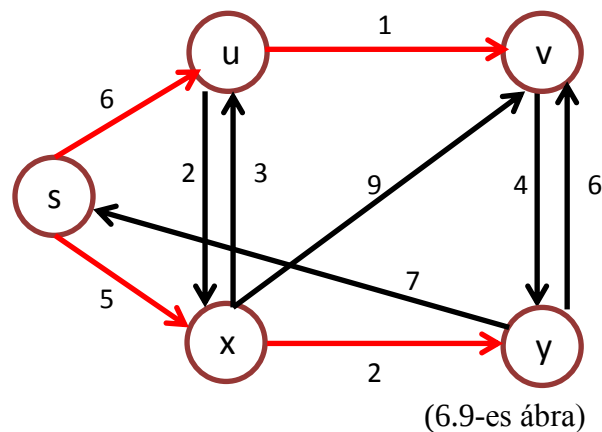
2. Oszlopok szerint haladunk a vizsgálat során, az s oszloppal kezdünk, mivel abból a csúcsból indulunk. A cellákban a vessző előtt álló szám jelenti az ebbe a csúcsba vezető út hosszát, a vessző jobb oldalán lévő írás arra utal, hogy melyik csúcsból jutunk ide. Például, az x-be vezető út hossza 5, az s csúcsból. Az s kezdőcsúcs esetében mindig „0,NIL”-t írunk. 0 jelenti, hogy ez a kezdőhely, NIL pedig hogy nem egy másik csúcsból jövünk. Az első sornak mindig így kell kinéznie, hogy a kezdőcsúcsnál „0,NIL”, a többi oszlopban pedig végtelen jel. A többi sorban a ∞ azt jelenti, hogy ebbe a csúcsba nem juthatunk el, abból, ahonnan most éppen vizsgálódunk.

Az 5 kisebb mint a 6, ezért az x csúcs a következő. Jön a táblázat harmadik sora, már nem nézzük azokat a csúcsokat, amikből már vizsgáldtunk, sem azt, amelyikben éppen vagyunk, ezért most azt kell megnézni, hogy az x -ből nézve mi a helyzet az y, u, v csúcsokkal. Az x -ből viszont már elérhető az y csúcs, ezért beírjuk, hogy „7,x” ($5+2$). Az x -nek az u szomszédja, viszont az u -ba már elmentünk, igaz hogy az s -ből, itt ez annyit jelent, hogy megint „6,s” kerül a cellába, mivel nincs ennél jobb választás. Aztán, az x -ből elérhető a v csúcs, az út hossza 9. Viszont ilyen helyzetben nem azt kell a cellába írni, hogy „9,x”, mivel az x -be az út hossza 5 volt, ezért ez összeadjuk, és „14,x” lesz.

A három érték közül a 6 a legkisebb, ezért a következő csúcsunk az u , és már nem nézzük az s és x pontokat. Az u -ból az y közvetlenül nem érhető el, ezért marad „7,x” a cellában. Az u -ból elérhető a v csúcs, amit mi előzőleg az x -ből közelítettünk meg. Az u - v távolsága 1, az u -ba viszont 6 egység az út, ezért „7,u” lesz a cellában.

Most a két szám ami a táblázat utolsó előtti sorában van egyenlő (a két 7-es), ezért válasszuk a balra lévőt, az y oszlopát. Már csak az y - v útról kell valamit mondanunk. Az y -ból a v elérhető lenne úgy, hogy $5+2+6=14$, de a „7,u” jobb választás, mivel a 7 kisebb mint a 14.

A táblázatban jelöljük valahogyan a számokat, amik magyarázzák a csúcsok kiválasztását (félkövér számok a táblázatban), majd rajzoljuk be a szükséges éleket a gráfunkba:



A Gale-Shapley-Algoritmus

A Gale-Shapley-algoritmus stabil házasságok meghatározására alkalmas.

Tétel: A Gale-Shapley algoritmus egy stabil házasságot eredményez.

Erről bőven elég ennyi, lássuk, hogy mire jó. Van egy jó példafeladat Tasnádi Attila: Számítástudomány gazdaságinformatikusoknak című tankönyvében.

Az urak és hölgyek rangsorai az alábbi két táblázattal adottak:

	<i>Feri</i>	<i>Jani</i>	<i>Pali</i>
1.	Kati	Kati	Juli
2.	Juli	Mari	Mari
3.	Mari	Juli	Kati

(6.6-os táblázat)

	<i>Juli</i>	<i>Kati</i>	<i>Mari</i>
1.	Pali	Feri	Feri
2.	Jani	Jani	Pali
3.	Feri	Pali	Jani

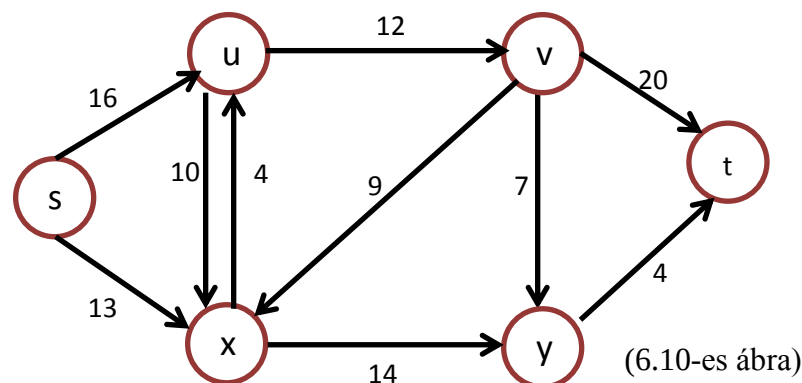
(6.7-es táblázat)

Egy lehetséges megoldás:

1. Feri eljegyzí Katit
2. Jani megkeresi Katit, aki elutasítja őt.
3. Jani eljegyzí Marit
4. Pali eljegyzí Julit
5. A Feri-Kati, Jani-Mari, Pali-Juli házasság adódik.

Ez az algoritmus erre jó, ne is essék több szó róla ☺

Hálózati folyamok



Ez egy hálózat. Az élek úgynevezett élkapacitással vannak ellátva, ami dióhéjban annyit jelent, hogy maximum annyi információ haladhat át az adott élen, amennyit a rajta lévő szám mutat. Az a cél, hogy az *s* csúcsból a *t* csúcsba a lehető legnagyobb mennyiséget eljuttassuk.

Definíció: adott egy $G = (V, E)$ irányított gráf, egy $s \in V$ forrás, egy $t \in V$ ($s \neq t$) nyelő, és egy $k : E \rightarrow [0, \infty)$ kapacitásfüggvény. A (G, s, t, k) négyes egy hálózat. Ebben a hálózatban az $f : E \rightarrow [0, \infty)$ egy s -ből t -be vezető folyam, ha

1. $f(e) \leq k(e)$ mindegyik $e \in E$ irányított élre
2. $\sum_{uv \in E} f(uv) = \sum_{vu \in E} f(vu)$ mindegyik $v \in V \setminus \{s, t\}$

Az $e \in E$ telítetlen, ha $f(e) < k(e)$, míg telített, ha $f(e) = k(e)$.

Vagyis a folyam egyetlen élen sem lehet nagyobb mint az élkapacitás. Az él akkor telítetlen, ha a rajta keresztülmenő folyamok értéke kisebb mint az élkapacitás, és akkor telített, ha már maximális ki van használva az él.

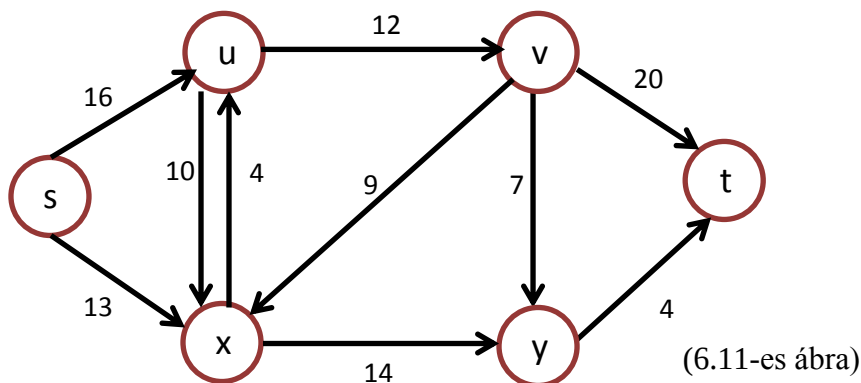
„Szemléltetésképpen feltehetjük, hogy a hálózattal egy vízvezetékrendszert ábrázolunk. A kapacitások a vezeték vastagságát jelentik, vagyis azt, hogy egységnyi idő alatt mennyi víz folyhat át azon a vezetékdarabon. A kérdés az, hogy egy adott hálózaton mennyi víz folyhat s -ből t -be. Szoktunk beszélni úthálózatról is, ahol a kapacitás az utak áteresztőképessége, és árukat kell eljuttatni a termelőtől a fogyasztóhoz” (Katona-Recski-Szabó: A Számítástudomány alapjai, 65. oldal)

„Legyen a gráfban $s = v_0, v_1, \dots, v_{k-1}, v_k = t$ egy út, aminek most nem kell feltétlenül az irányítás szerint haladnia. Növelhetjük a folyam értékét abban az esetben, ha minden $i = 0, 1, 2, \dots, k-1$ -re vagy $e_i = (v_i, v_{i+1})$ és $f(e_i) < c(e_i)$, vagy $e_i = (v_{i+1}, v_i)$ és $f(e_i) > 0$. Ekkor az első típusú éleken növeljük a folyam értékét, míg a második típusúakon csökkentjük, így az össz folyamérték nő. Az ilyen utakat javító útnak hívjuk.” (Katona-Recski-Szabó: A Számítástudomány alapjai, 65. oldal) $\rightarrow$ a példánál világossá válik (remélhetőleg ☺)

Tétel: Egy folyam értéke akkor és csak akkor maximális, a nincs javító út s -ből t -be.

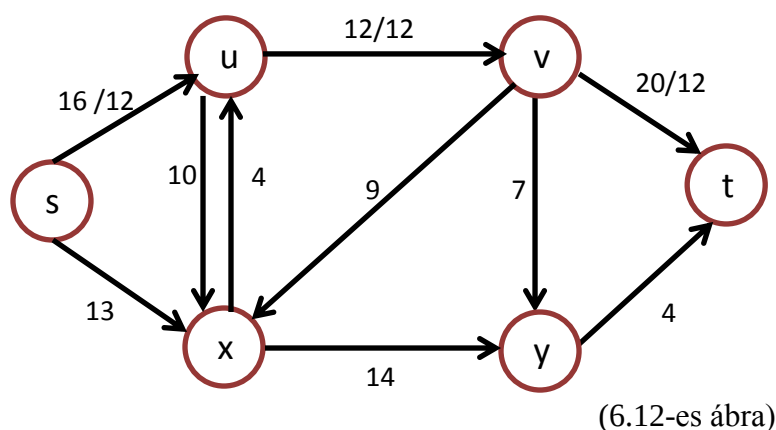
Ford-Fulkerson-tétel: Egy hálózatban a maximális folyam értéke megegyezik a minimális vágás értékével.

Most pedig térjünk vissza a fejezet elején lévő hálózatunkra.



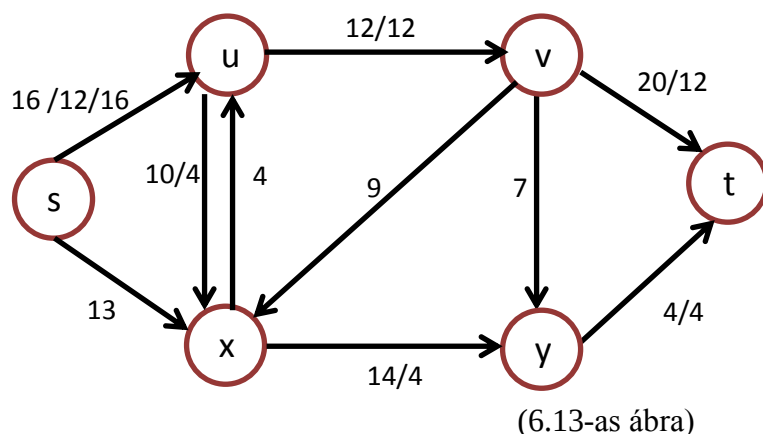
A feladatban a t csúcsba két él megy, egyiknek 4, a másiknak 20 az élkapacitása. Ez azt jelenti, hogy maximum $20 + 4 = 24$ értékű folyam mehet a nyelőbe. Kezdjük el az s csúcsból a t -be juttatni a folyam(okat) az éleken keresztül.

Nézzük először az $s \rightarrow u \rightarrow v \rightarrow t$ utat. Ezen az úton 3 él van, 16,12 és 20 élkapacitással. Ilyenkor a 3 mennyiség közül a legkisebb érvényes (12) az ezen az úton végigmenő folyam szempontjából (gondoljunk a vízvezetékes példára). Hogy csináltunk is valamit, jelöljük az ábrán, amit a következő képpen alakítunk át:



(6.12-es ábra)

Mint látjuk, az uv él már „használatlan”. Ebben a hálózatban már csak egyféle utat lehet választani, amin folyam mehet. Ez lehet az $s \rightarrow u \rightarrow x \rightarrow y \rightarrow t$, vagy pedig az $s \rightarrow x \rightarrow y \rightarrow t$, mindkettőn maximum 4 lehet az értéke, mivel az $y \rightarrow t$ élnek 4 az élkapacitása, és a lehetséges utakon ez a legkisebb érték. Válasszuk mondjuk a hosszabbat. Aztán meg kell adni a felhasznált javító utakat a teljes megoldáshoz, a végeredmény a következő:



(6.13-as ábra)

Felhasznált javító utak:

$s \rightarrow u \rightarrow x \rightarrow y \rightarrow t$ 4

$s \rightarrow u \rightarrow v \rightarrow t$ 12

A maximális folyam értéke 16. A Ford-Fulkerson tétel kimondja, hogy a maximális folyam értéke megegyezik a minimális vágás értékével.

Az $(\{s,u,x,y\}, \{v,t\})$ vágás értéke 16.

[Ugrás a feladatokhoz!](#)

VII. Rendezések

Ebben a fejezetben 5 féle rendező eljárásról lesz szó, ezek:

- *Beszűrő rendezés*
- *Összefésülő rendezés*
- *Kupacrendezés*
- *Gyorsrendezés*
- *Radix rendezés*

Mi ezzel az 5 féle eljárással fogunk foglalkozni. Mindegyiket egy-egy példafeladaton keresztül mutatjuk be.

Példa:

Rendezze a következő számsorozatot növekvő sorrendbe beszűrő-, összefésülő-, kupac-, majd gyorsrendezést alkalmazva!

5	2	4	6	1	3
---	---	---	---	---	---

Beszűrő rendezéssel:

Beszűrő rendezésnél a balról számított 2. számot nézzük először, ezt összehasonlítjuk az előtte lévő számmal. Ha a szám, amit vizsgálunk kisebb, mint ami előtte van, akkor azok helyet cserélnek.

Ezután a 3. helyen álló számot vizsgáljuk meg. Szintén összehasonlítgatjuk az előtte álló számokkal, amíg nem érkezünk olyanhoz, ami nála nagyobb. Ha ez megtörténik, akkor megállunk, és beírjuk az általunk vizsgált számot az elé, aminél az kisebb.

Ezután jön a 4. helyen álló szám, melyet összehasonlítunk az előtte álló 3-al, és az elé írjuk be, ami már kisebb nála. Ezt csináljuk addig, amíg végig nem értünk a számsorozaton. Nem nehéz kitalálni, hogy ha így teszünk, akkor minden szám pont a megfelelő helyre kerül, és az utolsó lépésben egy növekvő számsor lesz a szemünk előtt.

Tulajdonképpen annyit kell csinálni, hogy elindulunk a 2. helyen álló számtól, majd egyesével haladunk jobbra, de közben az éppen aktuális számot beírjuk a számtól balra lévő részen a helyére.

Tehát a megoldás:

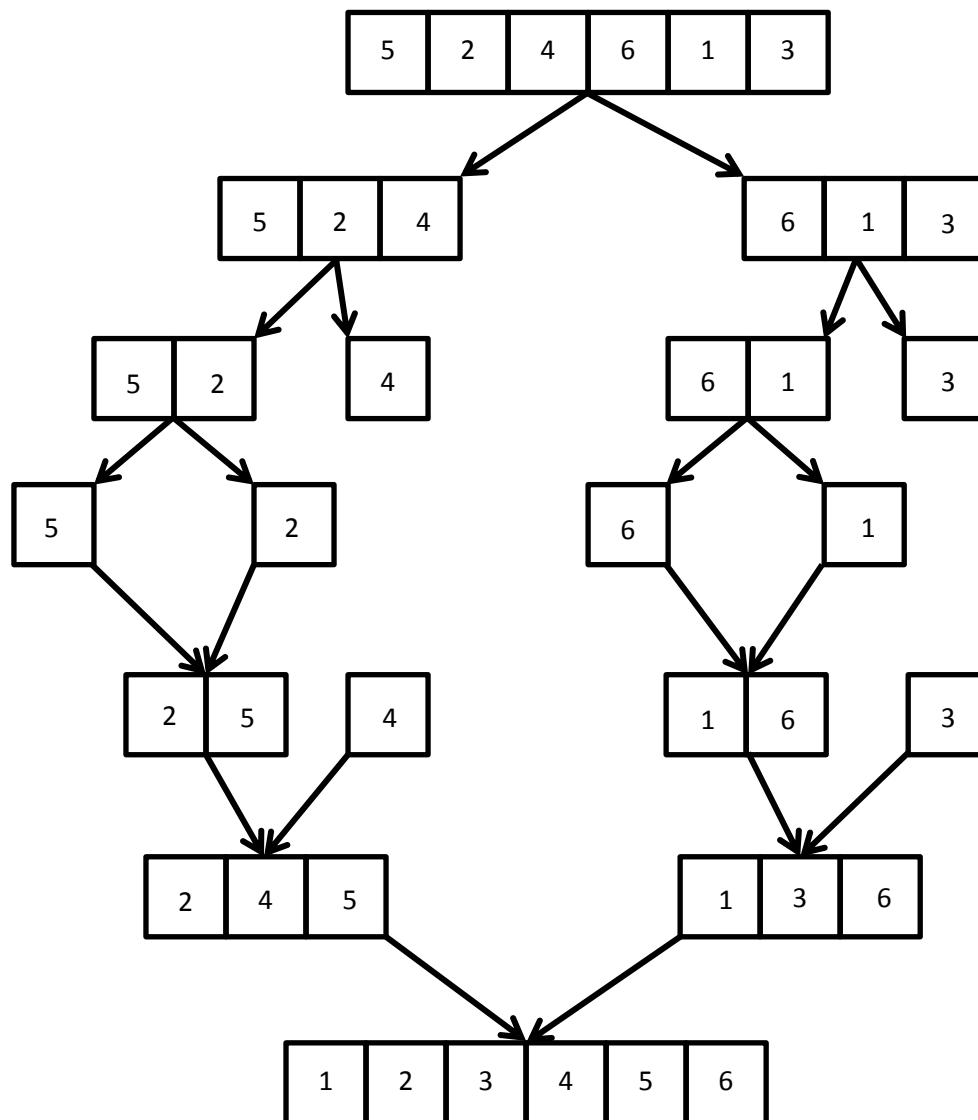
5	2	4	6	1	3
2	5	4	6	1	3
2	4	5	6	1	3
2	4	5	6	1	3
1	2	4	5	6	3
1	2	3	4	5	6

Ez az alap sorrend, és itt a 2. helyen lévő számot vizsgáljuk először, a kettést.

Az egyes lefutások után így alakul a számok sorrendje, és a piros színnel jelölt számokat vizsgáljuk.

(7.1-es táblázat)

Összefésülő rendezéssel:



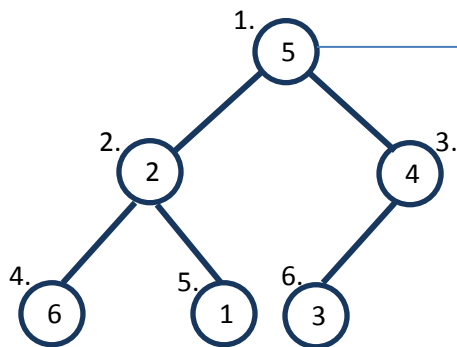
(7.1-es ábra)

Reméljük, hogy mindenki látja az ábrán, hogy hogyan működik, nagyon egyszerű. Az eredeti számsorozatot szétbontogatja felezéssel addig, amíg nem lesz minden számjegy

„egyedül”, majd szépen egyesíti lépésenként, már növekvő sorrendben. Annyit tennék hozzá, hogy amikor páratlan elemszámú tömböt bont szét az eljárás, akkor a páros számú tömb kerül a bal oldalra az ábrán, míg a páratlan elemszámú jobb. Ez látható 2. és a 3. sorban.

Kupacrendezéssel:

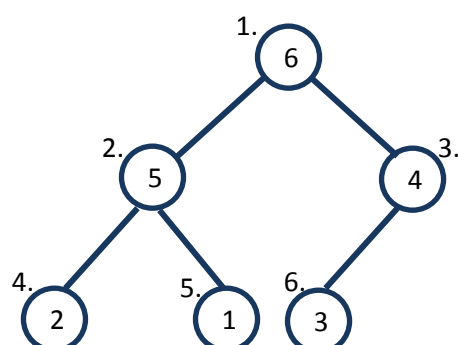
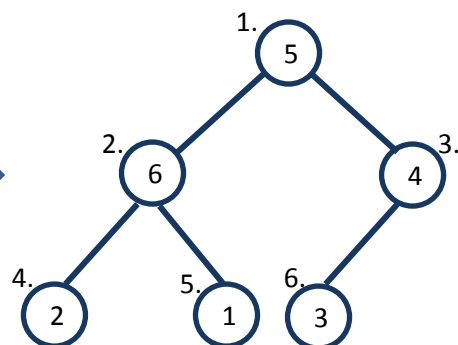
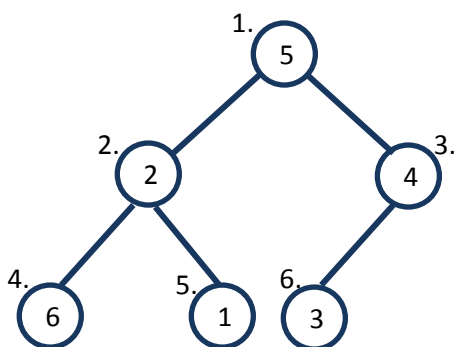
Ezt is főleg a rajz alapján érdemes megérteni, de előtte pár szót szölgünk róla általánosságban. A számokat először rendezetlenül beleírjuk az ábrán található megfelelő helyre, vagyis:



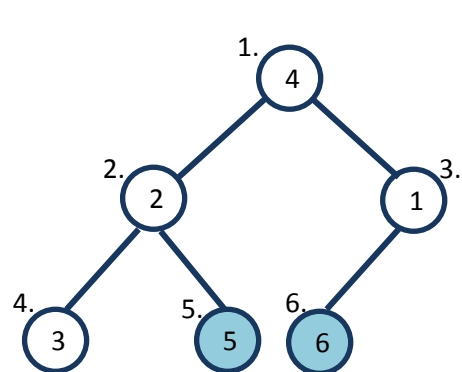
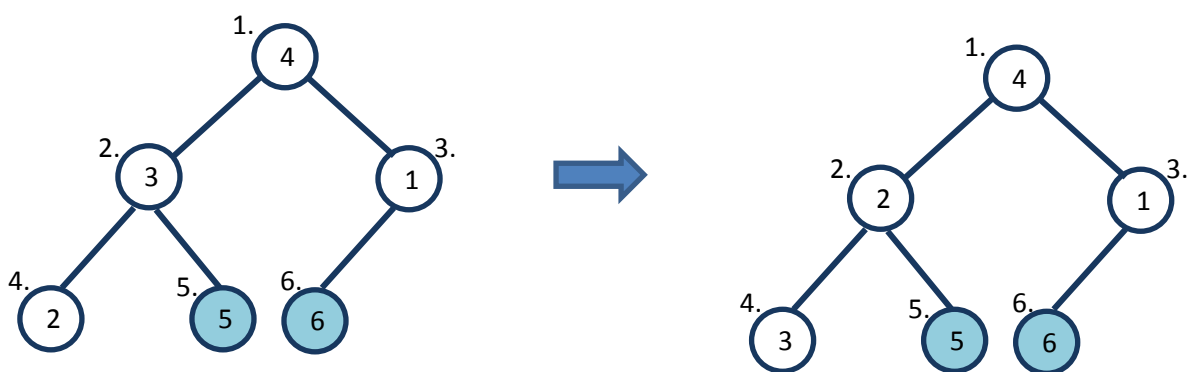
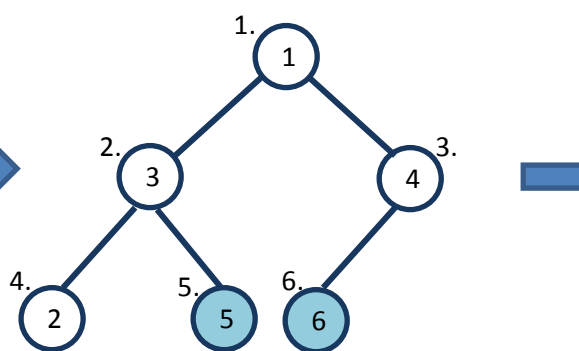
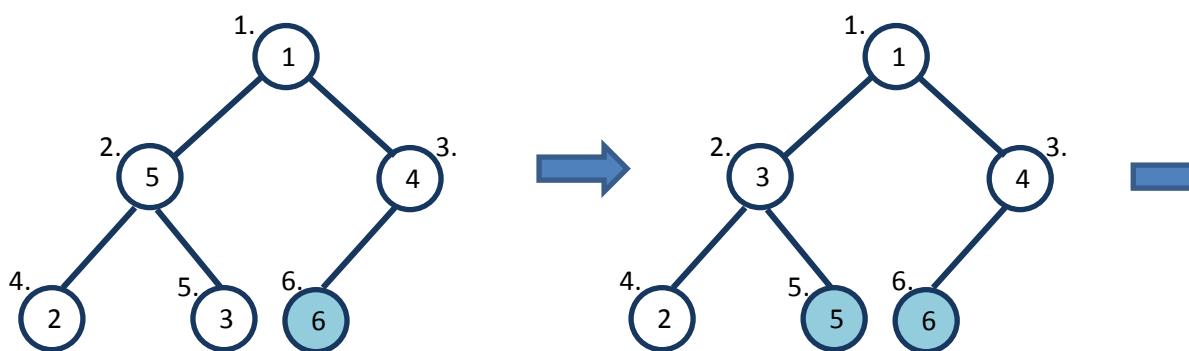
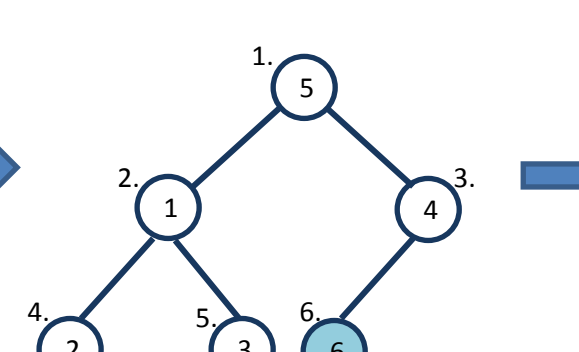
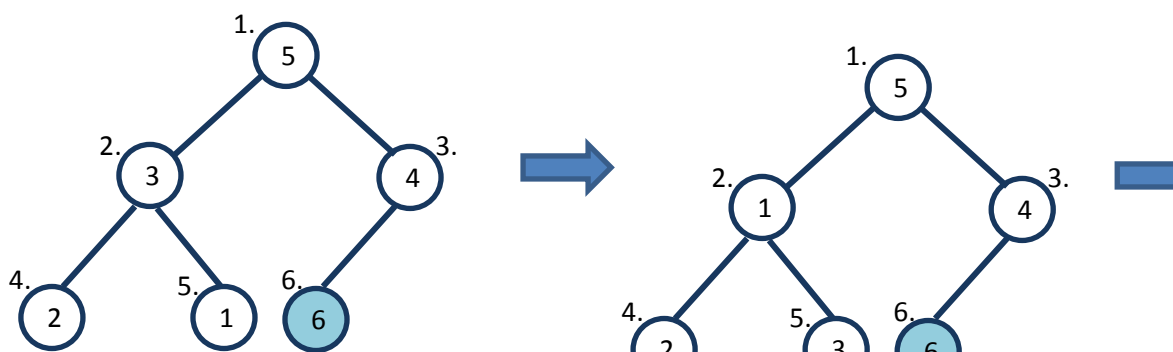
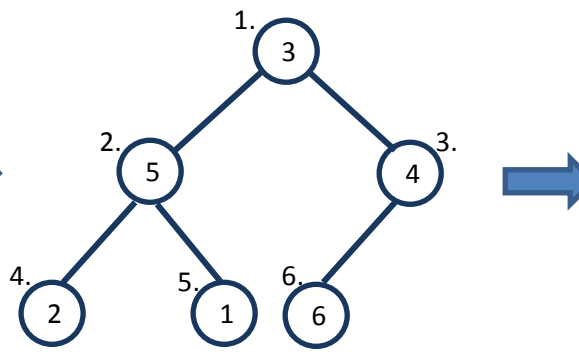
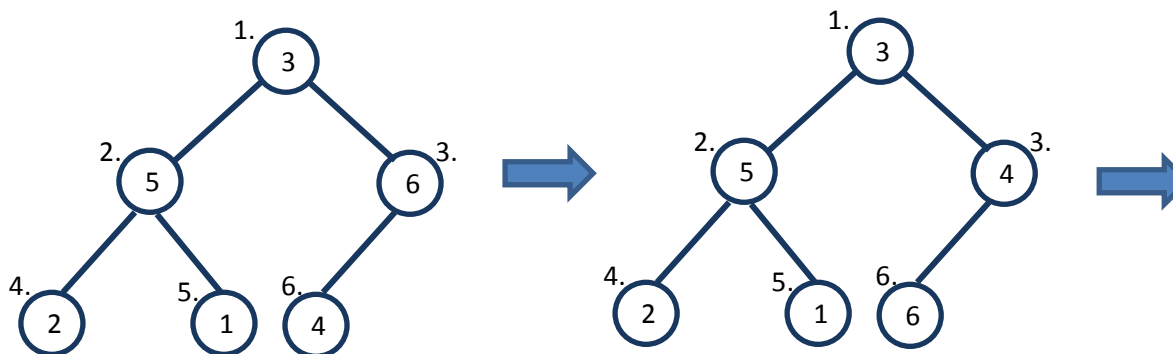
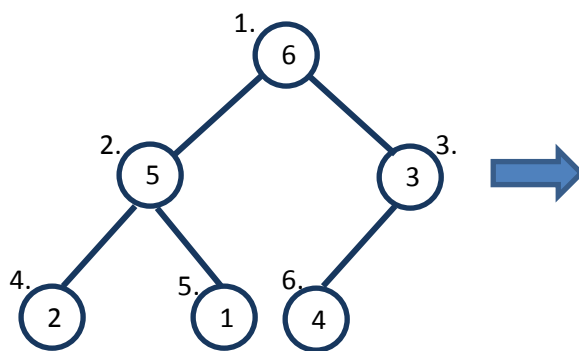
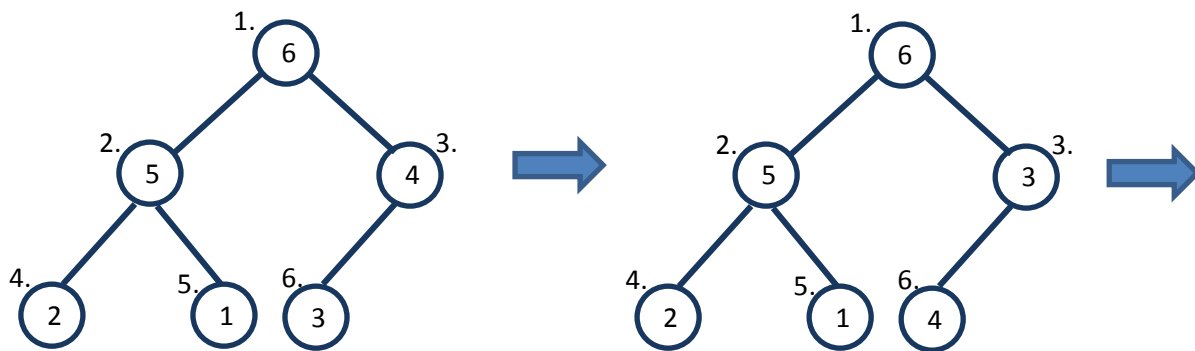
(7.2-es ábra)

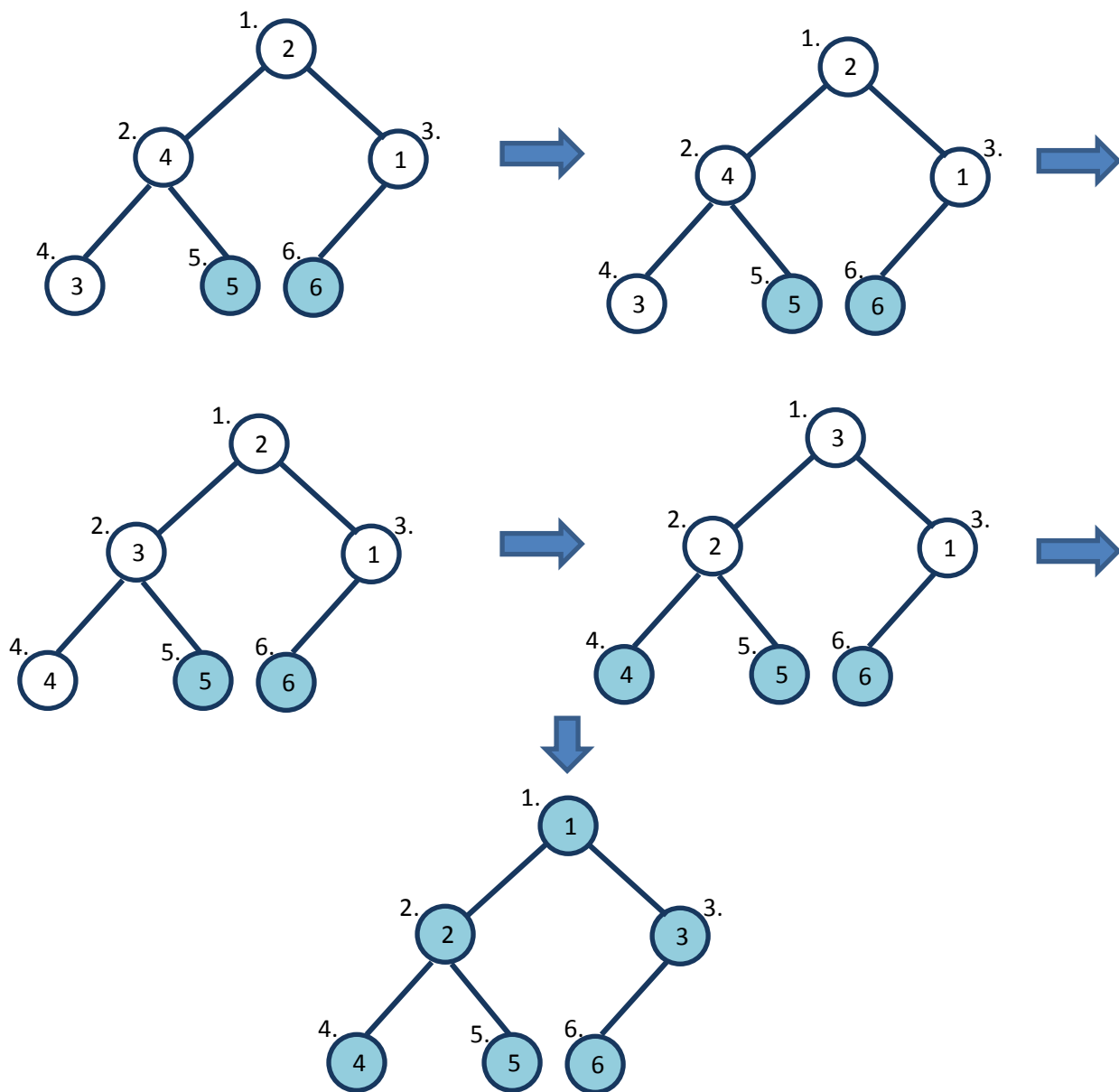
Szülő csúcsnak hívjuk, ami alatt helyezkednek el még csúcsok. Az 1. csúcs a 2. és a 3. csúcs szülője, tehát a 2. és a 3. csúcs az 1. csúcs gyermeke. Az 1. csúcs a fa gyökere.

Ahhoz, hogy ezzel az eljárással növekvő sorrendbe lehessen állítani a számokat, először úgynevezett kupacokat kell kialakítani. Akkor alakult ki a kupac, ha a szülők mindenhol nagyobbak a gyermekeiknél. A kupac kialakítását mindig alulról kezdjük el, tehát a következő képpen fog kialakulni:



Most, hogy minden szülő nagyobb a gyermekénél, kezdődhet a növekvő sorrendbe állítás. Meg fogja állapítani az algoritmus hogy melyik a legnagyobb szám, és azt helyre rakja, itt a 6. helyre fog kerülni. Utána megkeresi a 2. legnagyobb számot, az pedig az 5. helyre fog kerülni, és így tovább. Nézzük meg, hogy hogyan történik.





(7.3-as ábra)

Kupacrendezésnél, először beírjuk a számokat szintfolytonosan, ahogy a feladat megadja. Aztán kupaccá alakítjuk, úgy, hogy minden szülő nagyobb a gyermekénél. Miután ez megtörtént kezdetjük a növekvő sorrendbe rakást. Mi látjuk hogy melyik számnak hova kéne süllyednie, a legnagyobb a legnagyobb sorszámú helyre. A számok cserélegetését alulról kezdjük, arról a helyről, ahova süllyednie kell az aktuális számnak. Az ábrákon nagyon jól látszik, hogy ez a cserélegetés hogyan történik meg. Miután egy szám besüllyedt a helyére, azután valószínűleg megszűnik a kupactulajdonság (nem igaz már, hogy a szülők nagyobbak a gyermekeknél). Ezért ha az aktuális számunk a helyén van, helyre kell állítani a kupacot, mielőtt a következő számot elkezdenénk ugyanúgy a helyére rakni. Tanulmányozza az ábrákat mindenki. ☺

Gyorsrendezéssel:

Az 5,2,4,6,1,3 számsort kell rendezni még mindig, de most gyorsrendezéssel. Az eljárás először szétbontja az alaptömbünket két részre, úgy, hogy választ egy számot, és az annál kisebbeket a tömb első felébe rendezi, a töle nagyobb vagy egyenlőket pedig magától jobbra.

Ez a bizonyos nevezetes szám legyen a tömb első eleme, az 5. Ezután hátulról kezdünk el egyesével lépegetni a számsor eleje felé addig, amíg 5-nél kisebb számot nem találunk. A 3 kisebb nála, ez megvan. Most pedig a tömb elejétől kezdünk egyesével jobbra lépegetni, amíg találunk egy olyan számot ami nagyobb nála vagy egyenlő vele (az 5-ről beszélünk). Így az első körben saját magát találja meg úgymond, tehát az 5-t választottuk. Ezek után kicseréljük az 5-t a 3-al, tehát a (3,2,4,6,1,5) tömbhöz jutottunk.

Most hátulról megyünk megint, amíg 5-nél kisebb elemet találunk, ez az 1 lesz. Előlről pedig addig megyünk, amíg nagyobb/egyenlő számot találunk 5-nél, ez pedig csak a 6-os lehet. Ha ez megvan, az 1-es és a 6-os helyet cserél. Így megkapjuk a (3,2,4,1,6,5) tömböt.

Most már felbontható két résztömbre a tömbünk, mivel többször futnia ennek a résznek nincs értelme. Tehát a (3,2,4,1) és a (6,5) résztömbökhöz jutottunk. Egyértelmű, hogy az első 4 helyre az első tömb elemei fognak kerülni, az utolsó 2 helyre pedig a 2. tömb két eleme. Most külön-külön a két résztömbre kell ismételnünk az eljárást, amíg egyelemű tömböket nem kapunk.

A (3,2,4,1) –ben a 3-ast választjuk ki, amihez viszonyítunk (ez az y). Hátulról haladva az 1-es kisebb nála. Előlről pedig a 3-nál nagyobb/egyenlő rögtön a 3 (ilyenkor ugye saját magát választja), ezek helyet cserélnek, és a (1,2,4,3) –t kapjuk. A (6,5) -n is lefut a vizsgálat, akkor a két szám helyet cserél, és a (5,6) –t kapjuk.

Ezek után a (1,2,4,3) két részre bomlik, mivel a tömb első felében csak a 3-nál kisebb számok vannak, a második felében pedig a nagyobb egyenlők. Megkapjuk a (1,2) és a (4,3) résztömböket, a (5,6) -ból pedig kettő darab egyelemű tömb lesz, az (5) és a (6), így ezzel tovább nem foglalkozunk.

Ez így folytatódna tovább, de írásban csalhatunk, hiszen csak meg kell cserélni a számokat és kialakul a helyes sorrend. Szóval, létrejön 6 darab egy elemű tömb: (1) (2) (3) (4) (5) (6), és a sorrend növekvő.

Radix rendezés

Itt nem ezzel a 6 számmal dolgozunk, mert ez a rendezés másra való, többjegyű számok/dátumok rendezésére például. Egy példán keresztül tökéletesen érthetővé válik.

Példa: Rendezzük a következő dátumokat: 2002.03.24 ; 2004.04.24 ; 2002.03.01 ; 1999.11.11 ; 2012.08.04 ; 1999.01.11.

Radix rendezésnél először a leghátsó szám (itt a nap) szerint tesszük növekvő sorrendbe, majd egyet balra haladva a következő szerint lesz növekvő a sorrend (itt a hónapok), majd a legelső számjegy szerint állítjuk fel a növekvő sorrendet (itt az évek). Ugyan ez a helyzet ha 3 jegyű számokat rendezünk, ha több jegyűeket akkor pedig csak a lépésszám növekszik. Tehát a feladat:

Alap sorrend	1.	2.	3.
2002.03.24	2002.03.01	1999.01.01	1999.01.01
2004.04.24	1999.01.01	2002.03.01	1999.11.11
2002.03.01	2012.08.04	2002.03.24	2002.03.01
1999.11.11	1999.11.11	2003.04.24	2002.03.24
2012.08.04	2002.03.24	2012.08.04	2003.04.24
1999.01.01	2003.04.24	1999.11.11	2012.08.04

(7.2-es táblázat)

Ha két szám egyenlő, akkor az kerül előrébb, amelyik egyébként is feljebb volt a listában.

[Ugrás a feladatokhoz!](#)

VIII. Formális nyelvek és automaták

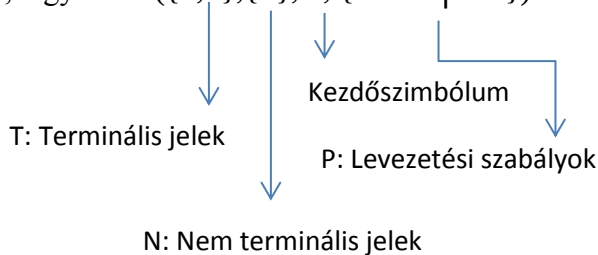
Ez talán a legnehezebb rész az egész tantárgyban, és nagyon – nagyon felszínesen fogunk belemenni, mindenképpen csak egy kiegészítő olvasmány funkciója van a rendes tankönyvekhez, előadásokhoz képest (mint az összes eddiginek, de ennek főleg). A rengeteg tétel és definíció közül itt csak azok fognak szerepelni, amik vagy nélkülözhetetlenek a feladatmegoldáshoz, vagy a minimum, hogy tudni kell. Nem teljesen az anyag elején kezdjük, de nem lesz gond.

Grammatika

Definíció: A $G = (T, N, S, P)$ egy grammatika, ahol

- T a terminális jelek véges nem üres halmaza. Ide tartoznak például a kisbetűk.
- N a nem terminális jelek, vagy más néven a nyelvtani jelek halmaza, melyre $T \cap N \neq \emptyset$. Ide tartoznak például a nagybetűk.
- S a kezdőszimbólum, ami egy nem terminális jel. Az S betűből fogjuk „kialakítani” a szavakat.
- P pedig a levezetési szabályok halmaza. Itt adjuk meg, hogy milyen jelet milyen jellel lehet átalakítani. $P \subseteq (T \cup N)^* N (T \cup N)^* \times (T \cup N)^*$.

Nézzünk meg egy grammatikát, legyen $G = (\{a, b\}, \{S\}, S, \{S \rightarrow ab \mid aSb\})$



Tehát, mindig az S kezdőszimbólumból indulunk. Azt mondják a levezetési szabályaink, hogy az S betű itt átírható ab -re, vagy aSb -re. Ha most ab -re írjuk át, akkor nincs több lehetőségünk, mivel azt már nem tudják mivé átalakítani. Ha aSb -vé alakítjuk, akkor mint látjuk megmarad középen egy S betű, amit szintén átírhatjuk ab -re, vagy aSb -re újra. Így kaphatunk $aabb$ vagy $aaSbb$ szavakat. Ezzel tovább tudunk játszani, és akár az $aaaaabbbbb$ szót is levezethetjük.

Azt mondjuk, hogy egyik szóból a másik közvetlenül (egy lépésben levezethető), ha a levezetési szabályok lehetővé teszik, hogy eljussunk egy szóból kívánt szóhoz. Ezt így

jelöljük: $\alpha \rightarrow \beta$. Például, a fent grammatikánál az aSb szóból az $aaSbb$ szó közvetlenül levezethető. Közvetett (legalább 2 lépést igénylő) levezetés alatt tulajdonképpen azt értjük, hogy egyik szótól a másikig úgy jutunk el, hogy kiírunk minden lépést, tehát például: $aSb \xrightarrow{G} aaSbb \xrightarrow{G} aaaSbbb \xrightarrow{G} aaaabbbb$.

Végül, a G grammatikában egy egyik szóból a másik szó levezethető, ha valamennyi lépés után eljuthatunk a kívánt szóhoz. Példa, jelölés: $aSb \xrightarrow{G}^* aaaaSbbbb$

Definíció: A $G = (T, N, S, P)$ grammatika által generált nyelv az

$$L(G) := \{x \in T^* \mid S \xrightarrow{G}^* x\}$$

Vagyis a nyelv az, ami olyan x szavakat tartalmaz, amik levezethetők az S kezdőszimbólumból.

Definíció: A G_1 és G_2 grammatikák ekvivalensek, ha $L(G_1) = L(G_2)$.

Például a $G_1 = (\{a\}, \{S\}, S, \{S \rightarrow a \mid aS\})$ és $G_2 = (\{a\}, \{S\}, S, \{S \rightarrow a \mid Sa\})$ grammatikák egyenlők, mivel mindkettő az $L(G_1) = L(G_2) = \{a^i \mid i \in \mathbb{Z}_+\}$ nyelvet generálja. Azért a^i , mivel amikor az S -t egy a betűvé írjuk át, bármikor is legyen ez a lépés, valahány számú a betű lesz egymás mellett. A hatvány azt jelenti, hogy mennyi a betű van egymás mellett. Tehát $a^5 = aaaaa$.

Tétel: Létezik grammatikával nem generálható nyelv.

Chomsky féle nyelvosztályok

Chomsky annak idején a generatív nyelvtanokat négy csoportba osztotta a felhasznált szabályok formája alapján. Ezek a következők:

- **0. típusú nyelvtanok** (rekurzíve felsorolható nyelvtanok)
- **1. típusú nyelvtanok** (környezetfüggő nyelvtanok)
- **2. típusú nyelvtanok** (környezetfüggetlen nyelvtanok)

A szabályok alakja lehet

$$A \rightarrow \alpha$$

Ahol A egy nem terminális jel, α pedig egy terminális és nem terminális jelekből álló szó. Akár üres szó is lehet. (Például $S \rightarrow ab \mid aSb$)

➤ **3. típusú nyelvtanok (reguláris nyelvtanok)**

A szabályok alakja lehet

$$A \rightarrow aB,$$

$$A \rightarrow a,$$

$$A \rightarrow \varepsilon$$

ahol A, B nem terminális jelek, a terminális szimbólum.

„Vegyük észre, hogy a 0. típus definíciója magába foglalja az összes többi, az 1. típus definíciója (legalábbis az (1.) leírás) 2. és 3. típust, a 2. típus definíciója a 3. típust. A nyelvtanokat tehát ez az osztályozás egymást sorban tartalmazó osztályokba sorolja, legszűkebb a 3., legtágabb a 0. osztály.

A nyelvtanok fenti osztályozása alapján a generatív nyelvek is osztályozhatók: i típusúnak mondunk egy nyelvet, ha generálható i típusú nyelvtannal. Később látni fogjuk, hogy hasonlóan a nyelvtanok osztályaihoz, a nyelvek is egymást sorban tartalmazó osztályokat alkotnak, legszűkebb a 3. típusú, azaz reguláris nyelvek osztálya, legtágabb a 0. típusú, azaz rekurzíve felsorolható nyelvek osztálya.” (...)

Ebből látható, hogy $\mathcal{G}_0 \supset \mathcal{G}_1 \supset \mathcal{G}_2 \supset \mathcal{G}_3$.

Chomsky féle hierarchia tétel: $\mathcal{L}_0 \supset \mathcal{L}_1 \supset \mathcal{L}_2 \supset \mathcal{L}_3$.

Véges determinisztikus automaták

Ebben a könyvben csak ezzel a fajta (a legegyszerűbb) automatával foglalkozunk, ezen kívül még létezik például veremautomata, Turing-gép, amik fontosak. Ezek megtalálhatók az órai előadásban, vagy bárhol máshol. ☺

Szükségünk lesz arra, hogy tudjunk az adott nyelvhez rajzolni egy azt felismerő automatát. Reguláris nyelvekhez fogunk véges determinisztikus automatát gyártani a későbbiek során.

Definíció: Az $\mathcal{A} = (N, T, M, S, F)$ ötös egy véges determinisztikus automata, ahol

- N a lehetséges állapotok véges nem üres halmaza; (a feladatokban az élekre kerülő kisbetűk, amik egyik csúcsból a másikba vezetnek)
- T az input ábécé, melyre $T \cap N = \emptyset$; (a nagybetűk, melyek a csúcsok lesznek az ábrán)
- M egy úgynevezett átmenetfüggvény, amely az $N \times T$ halmazból az N -be képező parciális függvény. (Ez fogja meghatározni, hogy melyik csúcsba milyen él megy. Tulajdonképpen azt mondja meg, hogy az ábrán melyik nagybetűből milyen kisbetű hatására megyünk át a másik nagybetűhöz. Ezért a mondatért szégyellem magam...)
- $S \in N$ a kezdőállapot
- $F \subseteq N$ a végállapotot halmaza.

Példa feladat: Legyen $L = \{(ab^2)^n \mid n \in \mathbb{N}^+\}$. Adjon meg egy L -t generáló 3-as típusú grammatikát, és rajzolja fel egy L -t felismerő véges determinisztikus automata átmenetdiagramját.

Megoldás: A feladat azt mondja, hogy a nyelvünk olyan szavakat generáljon, ami olyan szavakból áll, ahol az abb résszavakat állnak egymás mellett n -szer (ahányszor akarjuk). Például tudjon ilyen szavakat előállítani: $abbabbabbabb$.

Hogyan is álljunk neki kitalálni, hogy ez milyen átírási szabályok eredménye lesz?

Ami biztos, hogy az S -ből indulunk. Mivel a kívánt szavaink 'a' betűvel kezdődnek, ezért az S -t biztos, hogy olyanra kell átírni, amiben van 'a' betű. Azonban ha $S \rightarrow a$ szabályt adnánk meg, akkor itt meg is állna a dolog. Ezért írunk mellé egy nem terminális jelet is, hogy tudjuk folytatni a levezetést. Legyen $S \rightarrow aA$.

Most már tovább lehet fűzni a levezetést, ha az A -t is átírjuk valamire. Mivel az abb -ben a következő betű aminek jönnie kell, az a b . Tehát az A -t kell átírnunk b -re, plusz valamire, amiktől lehetővé válik, hogy azt is továbbfűzzük. Gondolom mindenki rájött, hogy a következő átírási szabályunk ez alapján az $A \rightarrow bB$ lesz.

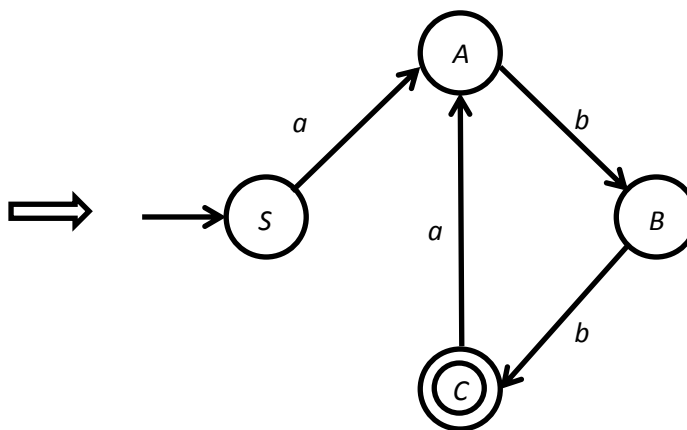
Nézzük az *abb* rész szót újra. Addig megvagyunk, hogy *ab*, és megint egy *b* betű következik. Ezt a következő féleképen lehet megoldani: a *B* nem terminális jelet írjuk át egyszer *b*-re. Így kapunk egy $B \rightarrow b$ szabályt, de ha ezt így hagyjuk akkor a nyelvünk csak egy darab *abb* szót tudna létrehozni. Hogy lehetne megoldani, hogy egymás után lehessen gyártani az *abb*-ket? Legyen az átírási szabályunk $B \rightarrow bC$.

Ezek után a következő szó válik levezethetővé: *abbC*. Nem nehéz észrevenni ezek után, hogy mit kell csinálni a *C*-vel. Legyen az átírás $C \rightarrow aA$. Most már képesek vagyunk az *abb*-t generálni annyiszor egymás mellé, ahányszor csak akarjuk, viszont van egy kis gond. A probléma pedig az, hogy nem állítható meg sehol sem a szavak létrehozása. Pont ezért kell a *C* átírását kiegészíteni erre: $C \rightarrow aA \mid \varepsilon$. Így már a nyelvünk rendesen működik, és mindig csak akkor tudjuk megállítani, amikor éppen egy *abb*-t leírt.

Tehát a grammatika: $G = (\{a,b,c\}; \{S, A, B, C\}; S; \{S \rightarrow aA, A \rightarrow bB, B \rightarrow bC, C \rightarrow aA \mid \varepsilon\})$.

Az automata:

Az átírási szabályokból rajzolhatók fel az irányított élek. A nagybetűkből (a csúcsok) a kisbetűk hatására, amiket az élekre írunk, megy át az automata következő csúcsba/csúcsokba.



(8.1-es ábra)

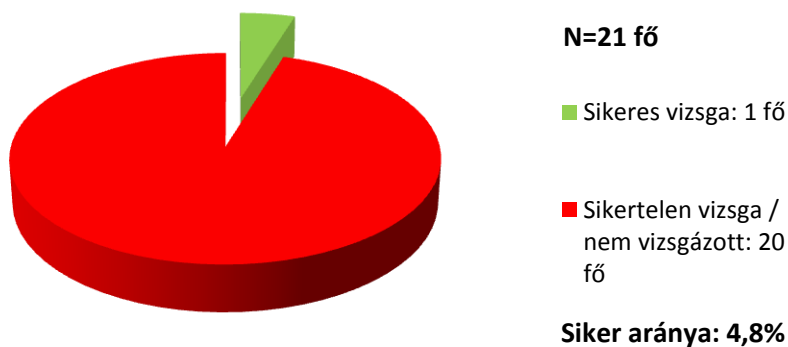
[Ugrás a feladatokhoz!](#)

IX. Utószó helyett

Primer kutatásunk során problémafelvetésként fogalmaztuk meg a gazdaságinformatikus hallgatók számítástudomány tantárgy elsajátításával kapcsolatos nehézségeit. Empirikus tapasztalat útján megállapítottuk, hogy a közös tanulás, a team-munka, a korrepetáció hallgatók közötti alkalmazása biztosította a sikeres elsajátítást. Így vetődött fel bennünk a gondolat, hogy mintegy „példatárat” alkotva „Puska helyett” készítsünk egy egyszerűsített nyelvű mégis problémamegoldást ösztönző, tömörített, a kor igényeihez igazodó példákkal tűzdelt, sokszínű és modern hangvételű módszertani összegzést.

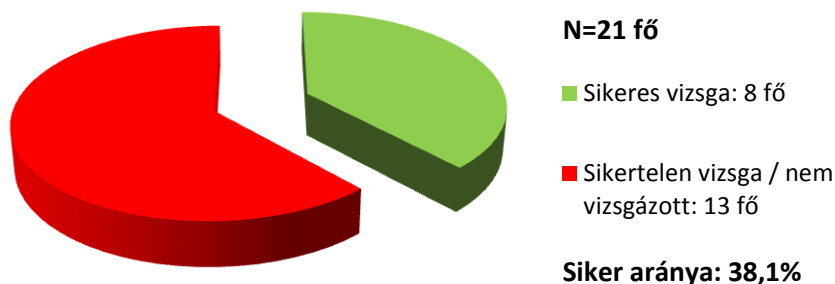
Kérdőívvel kívántuk felmérni, mintegy tesztelve a hallgatók hajlandóságát munkánk iránt. Most nézzük meg, hogy milyen hatása volt a hallgatókra a segédlet megjelenésének. Először diagramokkal szemléltetjük a vizsgák sikerességét. Az első ábra azt mutatja, ami „anyagunk” megjelenése előtti tanulási tendenciát, illetve modellt jellemezte, második pedig a tananyagunk alkalmazásnak sikerességére kívánja felhívni a figyelmet a múlt –és a jelen egymás alá helyezésével. Mindkét minta 21 fő teszterziókénti megkeresésére irányult, vagyis az adott évfolyamon tanuló összes gazdaságinformatikus hallgató megkérdezésére törekedtünk. (N=21)

A segédlet előtt



(9.1-es ábra)

A segédlet használatával



(9.2-es ábra)

Mint látjuk a sikerességi arány 4,8% volt, ami a segédanyag megjelenése után 38,1%-ra nőtt. Tehát a 2011/2012-es tanévben 21-ből 8 hallgató teljesítette a tantárgyat. A plusz 7 sikeres vizsga magyarázható lehetne esetleg azzal, hogy több idő volt felkészülni a dolgozatra, azonban tudnunk kell, hogy annak idején 7 fő kért tőlem korrepetációs segítséget (team munka). Annak idején az akkoriban szerkesztett még „félben lévő” „Puska helyett” című segédanyag alapján készülünk a kollokviumra. A vizsga után éreztük, hogy jobb eredmények születtek, ezért úgy gondoltuk, hogy érdemes mélyebben foglalkozni ezzel a témával, majd kérdőívet is készítettünk.

A „kiskönyvvel” kapcsolatosan kitöltött kérdőívek áttekintése

Ez 41 ember válaszait és véleményét dolgoztuk fel. Sok olyan kérdést tettünk fel, amiben a válaszadóknak pontozniuk kellett egy listán, 1-től 10-ig (Likert-skála), ahol, az egy a legkevésbé elégedett, míg a 10 a leginkább elégedett mértéket tükrözte. Nyitott kérdéseket is alkalmaztunk, melyekben a véleményüket kellett kifejezni a válaszadó megkérdezetteknek. Ezeket most röviden, összesítve mutatjuk be, majd kiemelünk pár előremutató véleményt, amit a kérdőívet kitöltők saját meglátásaik alapján osztottak meg velünk.

Abban a részben, amikor a hallgatókat arra kértük, hogy egy skálán pontozzák a segédlettel kapcsolatos kérdéseket, nagyon pozitív visszajelzéseket kaptunk. Egy felsorolásban szemléltetném, hogy a különböző szempontoknak hogyan felelt meg a kiskönyv, pontokban mérve, amit számtani átlaggal számítottunk ki és szemléltetni kívánunk a következők szerint:

- Ön a felkészülésben mennyire támaszkodik a Puska helyett című segédanyagra?

Eredmény: 7,5 pont

- Ön szerint van értelme a segédlet létezésének?

Eredmény: 8,5 pont

- Milyen mértékben könnyíti meg a segédanyag a felkészülést?

Eredmény: 9 pont

Az eredmény önmagáért beszél.

Megkértük a válaszadókat, hogy jellemezzék egy szóval a Számítástudomány tantárgyat. Erre a kérdésre rengetegféle válasz érkezett, a „felfoghatatlan”-tól a „kiihívás”-ig rengeteg alternatívát biztosítva számunkra, de még a „Chuck Norris” név is szerepelt listában.

Ezek után következtek a könyv szerkezetére, nyelvezetére és a benne található feladatokra vonatkozó kérdések következtek. Ezeket is megfelelőnek találták a válaszadók, a következőkhöz hasonló válaszokat osztottak meg velünk:

- *„Szerkezeti felépítése szinte kiváló, hisz el van benne magyarázva az anyag, ugyanakkor lehetőség van arra, hogy saját magunkat ellenőrizzük, az adott részek végén található feladatok megoldásával. Továbbá a jó felépítéshez még hozzátartozik, hogy a feladatok megoldásai nem rögtön a feladatok után vannak, hanem a segédanyag végén.”*
- *„nyelvezete érthető, nem nyers, hanem hallgató-centrikus, igazán könnyen befogadható”*

A feladatok számát többen kevesellték, de elégedettek voltak a meglévő példákkal. Szinte minden hallgatónak volt egyéni ötlete arra, hogy mit lehetne, kellene változtatni a segédlettel kapcsolatban. Ezt a későbbiekben pótolni kívánjuk, a tananyag fejlesztése hosszú távú célunk.

„Jó, hogy a különböző feladattípusokra van egy-egy példa, azonban véleményem szerint, ezeket még lehetne további feladatokkal bővíteni adott feladattípuson belül, hogy a hallgató be tudja gyakorolni az adott feladattípus megoldási vázát.”

Érezhetően jobb eredmények születtek, tehát mint a segédlet megjelenése előtt, az a tesztkérdőív egyúttal igényfelmérést is hivatott volt szolgálni. Összefoglaltuk a legfontosabb előremutató javaslatokat, megállapíthattuk tehát tézisként, hogy *„hasznos volt ennek a segédanyagnak az összeállítása, nem csak a saját felkészülésünk miatt, de segítséget nyújtott a számunkra, és a többiek számára is.”* Segítő szándékkal készítettük anyagunkat, melyet a többiek véleménye és a saját tapasztalatainak alapján továbbfejleszteni kívánunk, majd egy kiforrottabb formájában szeretnénk kiadni, esetlegesen elektronikus tananyagként felajánlani a főiskolának.

Melléklet

X. Feladatmegoldó Program

Mintegy kiegészítés képpen úgy gondoltuk, hogy jó lenne egy másik megoldás is, ami segítheti a vizsgákra való felkészülést. Azt gondoltuk ki, hogy írunk egy programot egy előre kiválasztott programozási nyelv segítségével, mellyel az adott témához kapcsolódóan lehetne feladatokat megoldani. Erre a célra a Visual Basic nevű programozási nyelvet választottuk. Ezen programozási nyelv korlátain belül szeretnénk megvalósítani ötletünket.

Elképzelésünk szerint a program egy főmenüvel kezdődne, amiből különböző „területekre” ugorhatunk, amik a következők:

- **Becslési feladatok megoldása:**
 - A felhasználó kiválaszthatja, hogy O vagy Ω egyenletet szeretne vizsgálni.
 - A felhasználó szabadon adhatja meg az egyenletekben konstans szorzószámokat és az n -ek kitevőit.
 - A program megállapítja, hogy helyes-e a felhasználó által megadott egyenlet, majd indoklást ad, hogy miért helyes vagy miért nem helyes.
- **Gráfalgoritmusok futtatása:**
 - A felhasználó kiválaszthatja, hogy milyen gráfon szeretné végrehajtani az eljárást.
 - A felhasználó megadhatja a gráfok élkapacitásait.
- **Rendező algoritmusok futtatása:**
 - A felhasználó szabadon adhat meg számokat, melyeket a tetszőleges algoritmus szerint sorba rendez a program.
- **Feladatmegoldás formális nyelvekre:**
 - Szeretnénk készíteni egy olyan ágot is a programba, amelyik képes megadni egy grammatika alapján a hozzá tartozó nyelvet, vagy a nyelv alapján a nyelvhez kapcsolódó grammatikát.

A programot úgy szeretnénk elkészíteni, hogy később fejleszthető legyen, esetleg új területekkel bővüljenek a feladatmegoldó részek.

XI. Példatár - Feladatok

Becslések

1. Igazolja, hogy $\log_2 n = O(n)$!
2. Igazolja, hogy $\log_2 n \neq \Theta(n)$!
3. Igazolja, vagy cáfolja, a $2^{n+1} = O(2^n)$!
4. Igazolja, vagy cáfolja, hogy $2^{2n} = O(2^n)$!

A következő feladatok megoldását egy az egyben az egyik forrásból vettük át, hozzászólás nélkül, így kicsit többet kell rajta gondolkodni, de nem baj ☺

5. Az alábbi függvényeket rendezze olyan sorozatba, hogy ha f_i -t f_j követi a sorban, akkor $f_i(n) = O(f_j(n))$ teljesüljön.
 - a. $f_1(n) = 11n^{2,5}$
 - b. $f_2(n) = \sqrt[5]{n} + 1000n$
 - c. $f_3(n) = 2^{\log^2 n}$
 - d. $f_4(n) = 2007n^2 \log n$
6. Igaz-e, hogy
 - a. ha $f = O(g)$ és $g = O(h)$, akkor $f = O(h)$?
 - b. ha $f = \Omega(g)$ és $g = \Omega(h)$, akkor $f = \Omega(h)$?

Az 1-4. feladatok forrása: *Tasnádi Attila: Számítástudomány Gazdaságinformatikusoknak, Bologna – Tankönyvsorozat, Budapest 2006.*

Az 5-6. feladatok forrása: *Friedl Katalin: Kiegészítő anyag az Algoritmelmélet tárgyhoz I. ; BME SZIT, 2008 május 22.*

[Ugrás a megoldásokhoz!](#)

Halmazelmélet, igazságtáblák és relációk

1. Írja fel az $A \Rightarrow B$ logikai műveletek diszjunktív normál formáját!
2. Igazolja igazságtábla segítségével, hogy $A \Leftrightarrow B = (A \Rightarrow B) \wedge (B \Rightarrow A)$!
3. Igazolja igazságtábla segítségével, hogy $A \Leftrightarrow B = (A \wedge B) \vee (\bar{A} \wedge \bar{B})$!
4. Igazolja igazságtábla segítségével, hogy $f(A,B,C) = (A \wedge B \wedge \bar{C}) \vee (\bar{A} \wedge B \wedge \bar{C}) \vee (\bar{A} \wedge \bar{B} \wedge \bar{C})$, ha $f(A,B,C) = „(0,1,0,0,0,1,0,1)”$.
5. Hozza egyszerűbb alakra a következő halmazalgebrai kifejezést, és az eredményt ellenőrizze igazságtáblával: $(A \cap B) \cup (A \setminus B) \cup (B \setminus A)$!
6. Hozza egyszerűbb alakra a következő halmazalgebrai kifejezést, és az eredményt ellenőrizze igazságtáblával: $A \setminus ((A \setminus B) \cup (A \setminus C))$!

7. Legyen $A = \{1,2,3,4,5,6\}$! Értelmezzük az $R \subseteq A \times A$ binér relációt úgy, hogy aRb akkor és csak akkor áll fenn $a,b \in A$ esetén, ha $(a-b)$ osztható 2-vel.
- R ekvivalencia reláció-e?
 - R parciális rendezés-e?
 - R teljes rendezés-e?
 - Ábrázolja a $G=(A,R)$ gráfot!

A válaszokat indokolja!

8. Legyen $A = \{2,4,6,8,10,12\}$! Értelmezzük az $R \subseteq A \times A$ binér relációt úgy, hogy aRb akkor és csak akkor áll fenn $a,b \in A$ esetén, ha $(a*b)$ osztható 6-al.

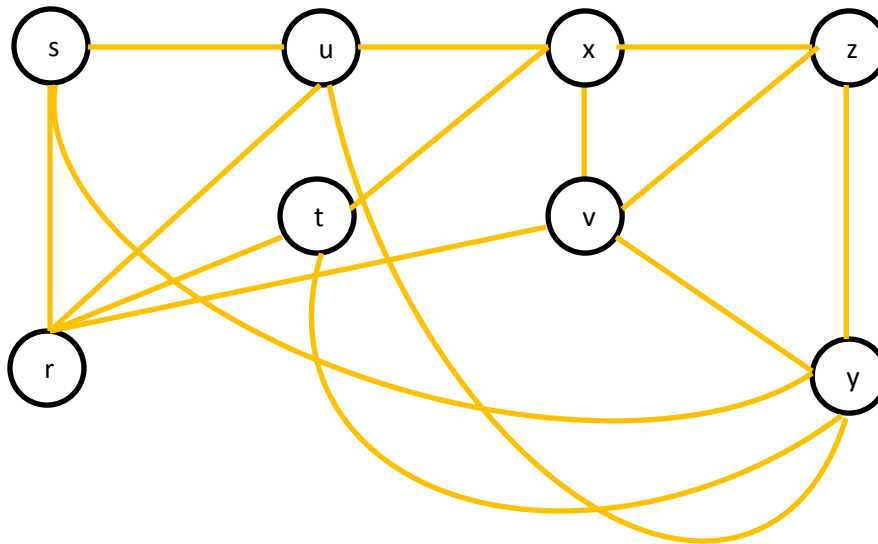
A homogén bináris relációk tulajdonságai közül melyekkel rendelkezik a relációnk?

A válaszokat indokolja!

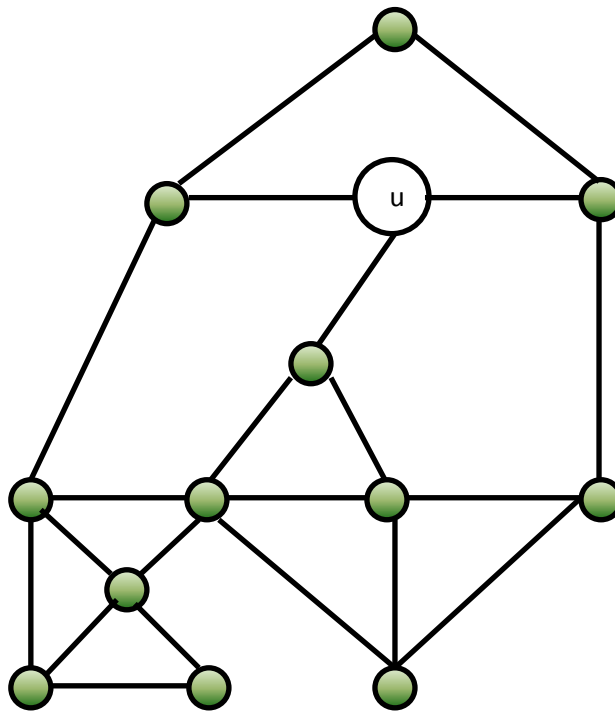
[Ugrás a megoldásokhoz!](#)

Gráfelmélet és Gráfalgoritmusok

1. Járja be a v csúcsból kiindulva szélességi és mélységi bejárás szerint a következő gráfot: (A feladat forrása: Tasnádi Attila: Számítástudomány gazdaságinformatikusoknak, 54. oldal, Bologna –Tankönyvsorozat, Aula Kiadó, 2006)

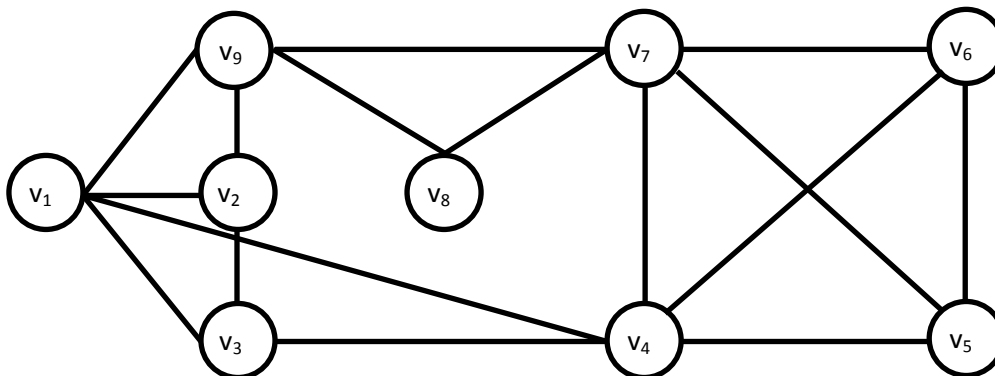


2. Járja be a gráfot szélességi, majd mélységi bejárás szerint az u csúcsból kiindulva!



(A feladat forrása: Tasnádi Attila: Számítástudomány gazdaságinformatikusoknak, 54. oldal, Bologna –Tankönyvsorozat, Aula Kiadó, 2006)

3. Járja be a gráfot szélességi, majd mélységi bejárás szerint a v_2 csúcsból kiindulva!



(A feladat forrása: Tasnádi Attila: Számítástudomány gazdaságinformatikusoknak, 54. oldal, Bologna – Tankönyvsorozat, Aula Kiadó, 2006)

4. A megadott szomszédsági lista alapján rajzolja fel a gráfot, majd járja be a gráfot az 1-es csúcsból szélességi, illetve mélységi bejárással. Irányított-e a gráf? A csúcsokból induló éleket a szomszédsági listában való előfordulás sorrendjében vegye figyelembe. Adja meg a gráfot szomszédsági mátrixszal is.

$1 \rightarrow 2 \rightarrow 4 \rightarrow 6/$

$2 \rightarrow 6/$

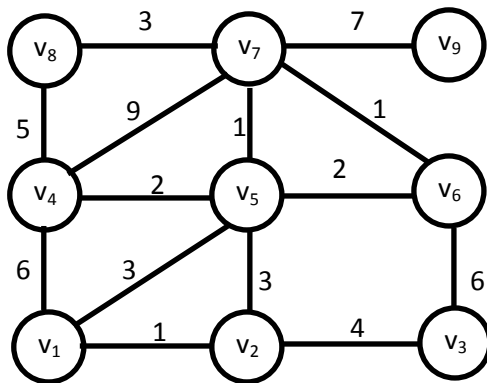
$3 \rightarrow 2/$

$4 \rightarrow 2 \rightarrow 5 \rightarrow 3/$

5 → 2 → 6/

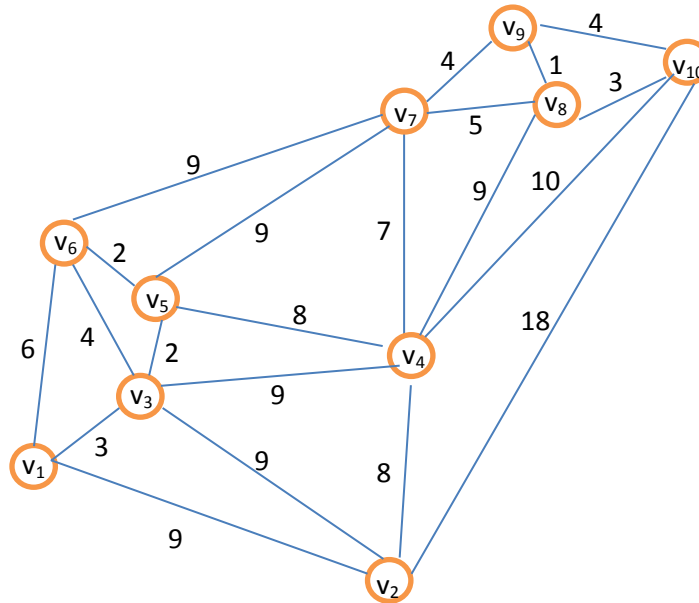
6/

5. Határozza meg a minimális feszítőfát Kruskal algoritmusával! Jelölje a gráfon a kiválasztott éleket, és dokumentálja a kiválasztásuk sorrendjét.

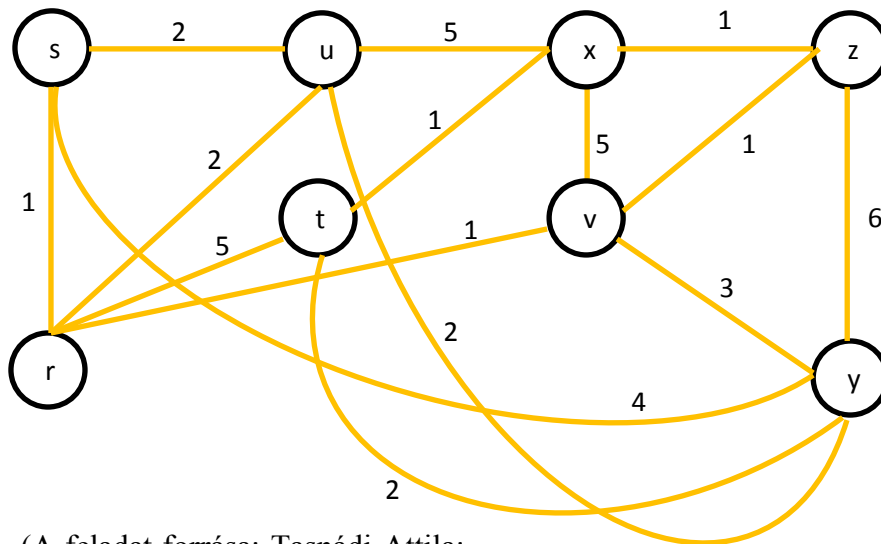


(A feladat forrása: Tasnádi Attila:
Számítástudomány
gazdaságinformatikusoknak, 56.
oldal, Bologna –Tankönyvsorozat,
Aula Kiadó, 2006)

6. Határozza meg a minimális feszítőfát Kruskal algoritmusával! Jelölje a gráfon a kiválasztott éleket, és dokumentálja a kiválasztásuk sorrendjét.

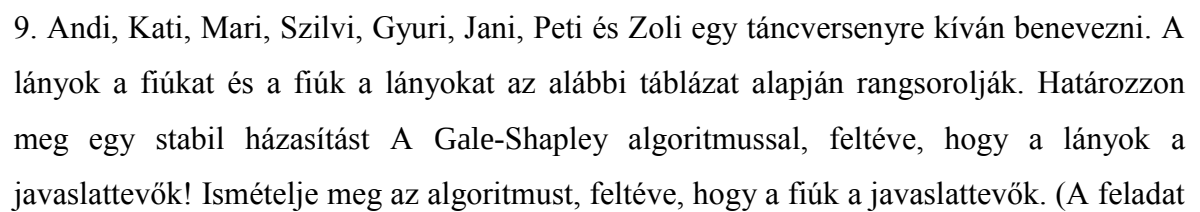


7. Határozza meg az alábbi gráfban az r csúcsból a gráf összes többi csúcsába vezető legrövidebb utak hosszait Dijkstra algoritmusával! A lépéseket dokumentálja táblázatos formában.



(A feladat forrása: Tasnádi Attila:
Számítástudomány
gazdaságinformatikusoknak, 56.
oldal, Bologna –Tankönyvsorozat,
Aula Kiadó, 2006)

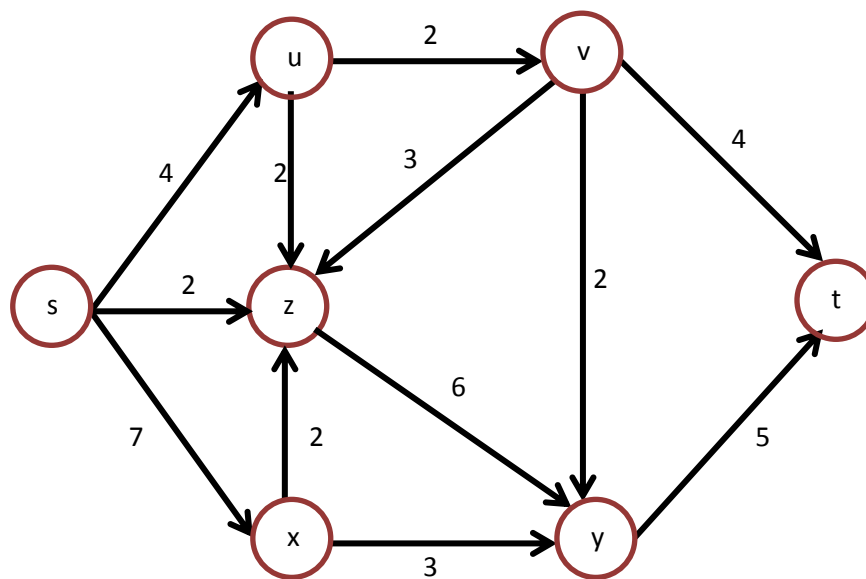
8. Határozza meg az a csúcsból az összes többi csúcsba vezető legrövidebb út hosszát Dijkstra algoritmusával! A lépéseket dokumentálja táblázatos formában.



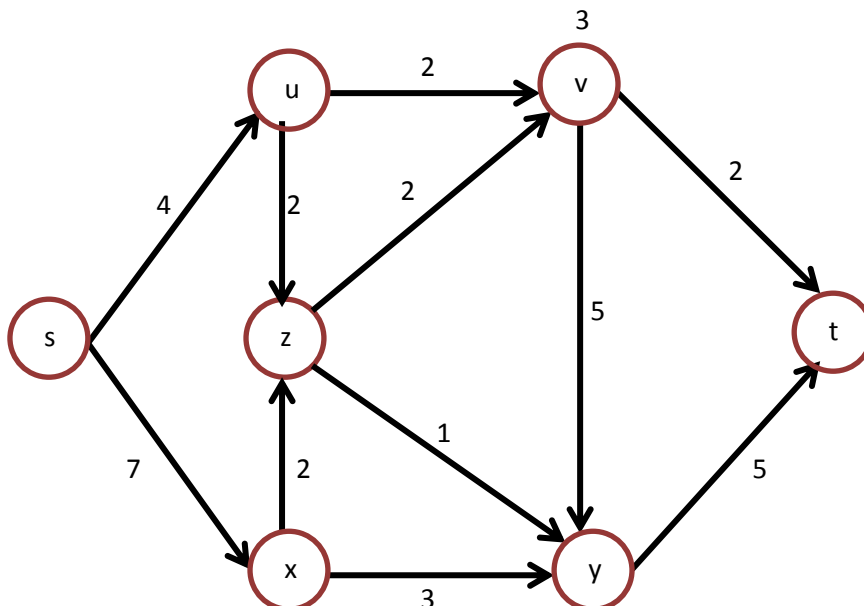
forrása: Tasnádi Attila: Számítástudomány Gazdaságinformatikusoknak, Bologna – Tankönyvsorozat, 2006, 56. oldal)

Helyezés	Andi	Kati	Mari	Szilvi	Gyuri	Jani	Peti	Zoli
1.	Zoli	Peti	Zoli	Peti	Andi	Mari	Kati	Mari
2.	Jani	Zoli	Peti	Gyuri	Kati	Kati	Andi	Szilvi
3.	Peti	Gyuri	Jani	Zoli	Szilvi	Szilvi	Szilvi	Kati
4.	Gyuri	Jani	Gyuri	Jani	Mari	Andi	Mari	Andi

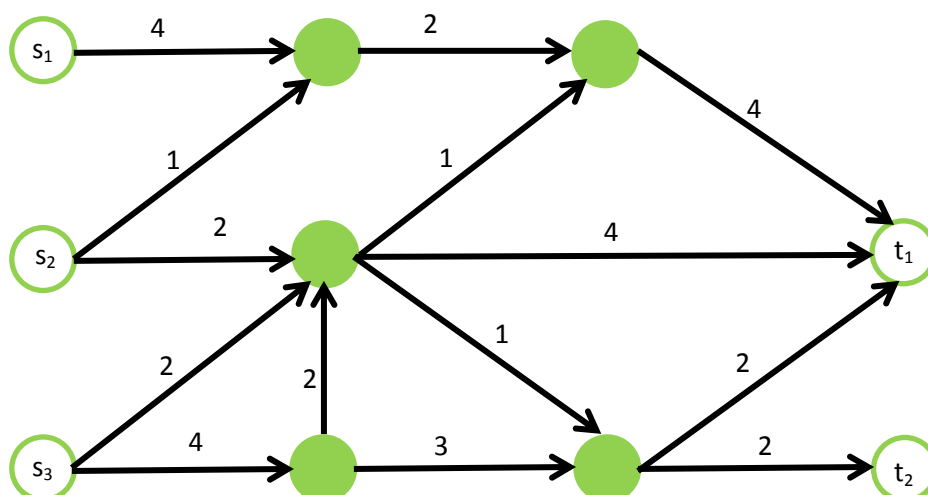
10. Határozza meg a maximális folyamat és annak értékét az s csúcsból a t csúcsba a 0 folyamból kiindulva! Írja le a felhasznált javító utakat! Vágás segítségével igazolja, hogy a kapott folyam valóban maximális. (A feladat forrása: Tasnádi Attila: Számítástudomány gazdaságinformatikusoknak, 56. oldal, Bologna –Tankönyvsorozat, Aula Kiadó, 2006)



11. Határozza meg az alábbi csúskapacitással is rendelkező hálózatbeli maximális folyam értékét! (Ötlet: a csúskapacitással ellátott csúcsok helyét vezessünk be két csúcsot.) (A feladat forrása: Tasnádi Attila: Számítástudomány Gazdaságinformatikusoknak, Bologna – Tankönyvsorozat, 2006, 57. oldal)



12. Határozza meg az alábbi ábrán látható háromforrású és kétnyelős kiterjesztett hálózatbeli maximális folyam értékét! (Ötlet: a források és a nyelők helyett vezessünk be egy-egy mesterséges csúcsot.) (A feladat forrása: Tasnádi Attila: Számítástudomány Gazdaságinformatikusoknak, Bologna – Tankönyvsorozat, 2006, 57. oldal)



[Ugrás a megoldásokhoz!](#)

Rendezések

1. Állítsa növekvő sorrendbe beszűrő rendezéssel a következő számokat: 6,2,1,5,7,4,2.
2. Összefésülő rendezéssel állítsa növekvő sorrendbe a következő számokat: 9,1,3,4,6,5,2,7,5.
3. Kupacrendezést alkalmazva állítsa növekvő sorrendbe a következő számokat: 16,14,10,8,7,9,3,2,4,1.
4. Állítsa növekvő sorrendbe gyorsrendezéssel a következő számokat: 5,3,1,8,9,4,3.
5. Radix rendezés segítségével rakja növekvő sorrendbe a következő
 - a. számokat: 329 ; 457 ; 657 ; 839 ; 436 ; 720 ; 355.
 - b. dátumokat: 1995.06.07 ; 1996.03.12 ; 1994.03.28 ; 1993.11.07 ; 1995.05.12 ; 1997.06.28 ; 1994.10.15 ; 1995.11.12 ; 1994.12.07.
6. Legyenek a tömb elemei 16,14,10,8,7,9,3,2,4,1. Rendezze a tömbelemeket növekvő sorrendbe
 - a. beszűrő rendezéssel
 - b. összefésülő rendezéssel
 - c. gyorsrendezéssel!

[Ugrás a megoldásokhoz!](#)

Formális nyelvek és automaták

1. Adjon meg olyan környezetfüggetlen grammatikát, ami a következő nyelvet generálja:
 $L = \{a^i b^i \mid i \in \mathbb{N}^+\}$.
2. Adjon meg olyan környezetfüggő nyelvtant, ami az $L = \{a^i b^j c^i \mid i \in \mathbb{N}^+\}$ nyelvet generálja!
3. Adjon meg olyan reguláris grammatikát, mely az $L = \{a^i b^j \mid i, j > 0\}$ nyelvet generálja!
4. Adjon meg olyan reguláris nyelvtant, ami az $L = \{a^i b^j c^k \mid i, j, k \in \mathbb{N}^+\}$ nyelvet generálja!
Készítsen hozzá véges determinisztikus automatát!
5. Legyen $G := (\{a, b\}, \{S\}, S, \{S \rightarrow aa, S \rightarrow bb, S \rightarrow aSa, S \rightarrow bSb\})$. Hányadik típusú a G nyelvtan? Határozza meg a G által generált nyelvet! Adja meg közvetlen levezetési lépések sorozatával az *abaaba* szó levezetését!

6. Legyen $L = \{(a^2b^3)^i \mid i \in \mathbb{N}^+\}$. Adjón meg egy L -t generáló 3-típusú grammatikát, és egy L -t felismerő véges determinisztikus automata átmenetdiagramját!
7. Legyen $L = \{x^{-1}x \mid x \in \{a,b\}^*\}$. Adjón meg egy L -t generáló 2-típusú grammatikát!

[Ugrás a megoldásokhoz!](#)

XII. Példatár - Megoldások

Becslések

1. Igazolja, hogy $\log_2 n = O(n)$!

Az összefüggés akkor lehet igaz, ha a $\log_2 n \leq c \cdot n$ megvalósul, tehát keresnünk kell egy megfelelő pozitív c értéket. Most egy kis trükk következik: minden pozitív n -re igaz, hogy $\log_2 n \leq 1 + n$, aki nem hiszi, járjon utána. ☺ Azért $1 + n$, mert így könnyebb lesz bizonyítani. Ha ez megvan, felírhatjuk, hogy $1 + n \leq 2n$ (ha $n \geq 1$). Összefoglalva az eddigieket: $\log_2 n \leq 1 + n \leq 2n$. Ebből már talán látszik, hogy a $\log_2 n \leq c \cdot n$ összefüggésnek eleget tesz a $c = 2$ konstans, tehát $\log_2 n = O(n)$ valóban igaz.

2. Igazolja, hogy $\log_2 n \neq \Theta(n)$!

Az előző feladatban láthattuk, hogy a $\log_2 n = O(n)$ teljesül, itt viszont azt látjuk, hogy a $\log_2 n \neq \Theta(n)$ összefüggés valami miatt nem igaz. Gyanakodhatunk, hogy ez azért van így, mert valószínűleg $\log_2 n \neq \Omega(n)$.

Hogy $\log_2 n = \Omega(n)$ fennálljon, szükség lenne egy olyan c pozitív valós konstansra, és egy olyan n_0 pozitív egészre (ugye a küszöbszám, amitől kezdve nézzük az n -eket... $n \geq n_0$!), hogy a $c \cdot n \leq \log_2 n$ teljesüljön. Azt kell mondjam, hogy analízisbeli ismereteink miatt kéne tudnunk erre a választ, mégpedig hogy ilyen c érték nem létezik, mivel $\lim_{n \rightarrow \infty} \frac{\log_2 n}{n} = 0$. Tehát $\log_2 n \neq \Omega(n)$, így $\log_2 n \neq \Theta(n)$.

3. Igazolja, vagy cáfolja, a $2^{n+1} = O(2^n)$!

A 2^{n+1} -ről tudjuk, hogy átírható más alakra, tehát $2^{n+1} = 2 \cdot 2^n$. Most meg kell vizsgálni, hogy a $2^{n+1} \leq c \cdot 2^n$ egyenlőtlenség fennáll-e valamilyen c érték esetén. Ez az egyenlőtlenség viszont teljesül, ha a $c=2$ (de a $c=3$ esetén is teljesülne). Nagy n -ekre tehát teljesül $2^{n+1} = O(2^n)$ egyenlőség, ha $c=2$.

4. Igazolja, vagy cáfolja, hogy $2^{2n} = O(2^n)$!

Alakítsuk át a 2^{2n} -t $2^n \cdot 2^n$ alakúra. Így már látható, hogy semmilyen c esetén sem teljesül, hogy $2^{2n} \leq c \cdot 2^n$. Tehát $2^{2n} \neq O(2^n)$.

5. Az alábbi függvényeket rendezze olyan sorba, hogy f_i -t f_j követi a sorban, akkor $f_i(n) = O(f_j(n))$ teljesüljön

$f_1(n) = 11n^{2.5}$	$f_2(n) = 5\sqrt{n} + 1000n$
$f_3(n) = 2^{\log^2 n}$	$f_4(n) = 2007n^2 \log n$

megoldás : $f_2(n) = 5\sqrt{n} + 1000n \leq 1005n$, ha $n \geq 1$ és ha $n > 2$, akkor $1005n \leq 2007n^2 \log n$, ezért $f_2 = O(f_4)$.

mivel $\log n \leq \sqrt{n}$, ezért $2007n^2 \log n \leq 2007n^2 \sqrt{n} = \frac{2007}{11} (11n^{2.5})$, tehát $f_4 = O(f_1)$

Írjuk át f_3 - at: $2^{\log^2 n} = 2^{\log n \cdot \log n} = (2^{\log n})^{\log n} = n^{\log n}$. Ha $n > 6$, akkor $\log n > 2,5$, és ezért ilyenkor $f_1(n) = 11n^{2.5} \leq 11n^{\log n} = 11f_3(n)$.

tehát a sorrend f_2, f_4, f_1, f_3

6 Igaz-e, hogy

(a) ha $f = O(g)$ és $g = O(h)$, akkor $f = O(h)$?

(b) ha $f = \Omega(g)$ és $g = \Omega(h)$, akkor $f = \Omega(h)$?

Megoldás:

(a) A feltételek szerint létezik olyan $c_1, c_2 > 0, n_0, n_1$ számok, hogy

$|f(n)| \leq c_1 |g(n)|$ ha $n \geq n_0$ és $|g(n)| \leq c_2 |h(n)|$ ha $n \geq n_1$.

Ezeket összerakva kapjuk, hogy $|f(n)| \leq c_1 |g(n)| \leq c_1 c_2 |h(n)|$ ha $n \geq \max(n_1, n_2)$,

tehát az állítás igaz.

(b) Az előzőhöz hasonlóan látszik, hogy ez is igaz.

Halmazelmélet, Rendezések, Relációk

1. Írja fel az $A \Rightarrow B$ logikai műveletek diszjunktív normál formáját!

$$A \Rightarrow B = \bar{A} \vee B \quad \text{vagy} \quad (A \wedge B) \vee (\bar{A} \wedge B) \vee (\bar{A} \wedge \bar{B})$$

2. Igazolja igazságtábla segítségével $A \Leftrightarrow B = (A \Rightarrow B) \wedge (B \Rightarrow A)$!

A	1	1	0	0
B	1	0	1	0
$A \Rightarrow B$	1	0	1	1
$B \Rightarrow A$	1	1	0	1
$(A \Rightarrow B) \wedge (B \Rightarrow A)$	1	0	0	1
$A \Leftrightarrow B$	1	0	0	1

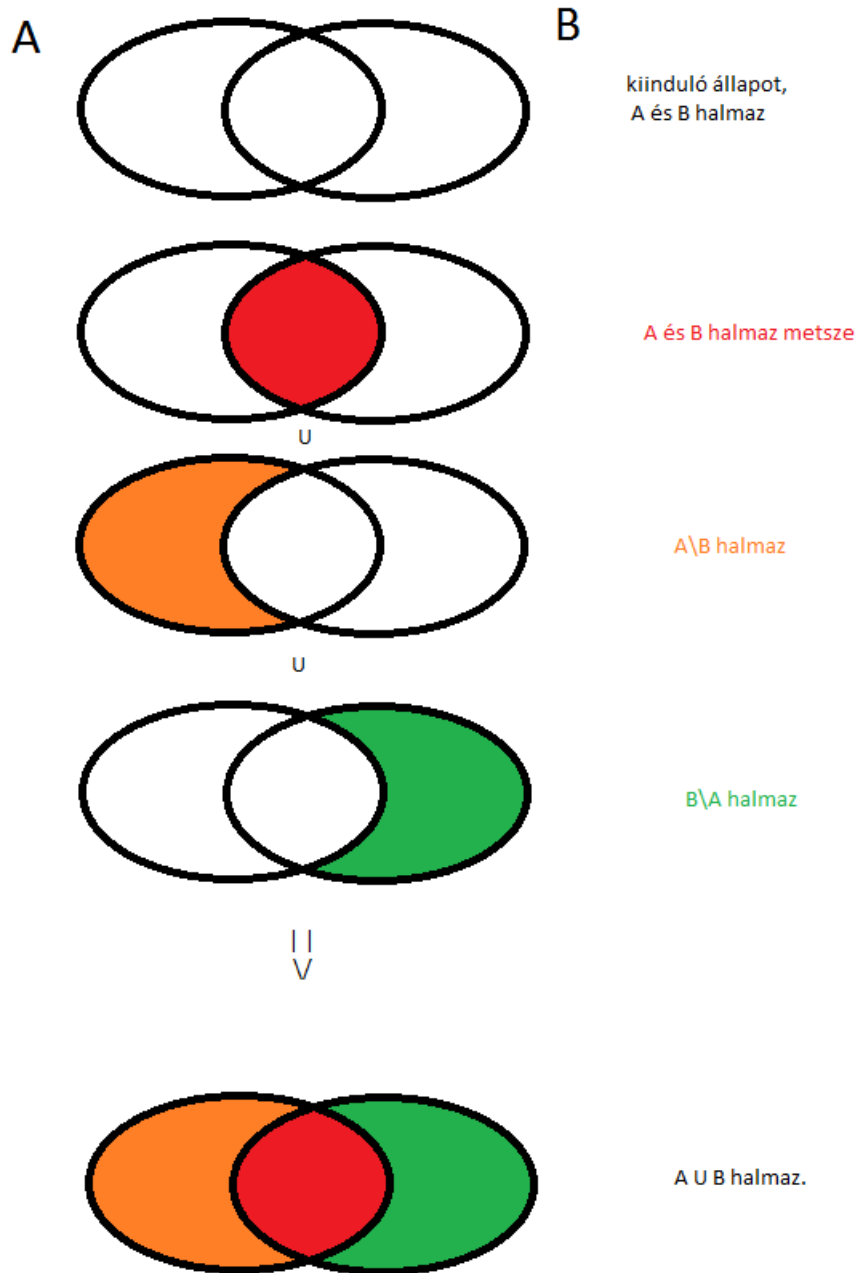
3. Igazolja igazságtábla segítségével $A \Leftrightarrow B = (A \wedge B) \vee (\bar{A} \wedge \bar{B})$!

A	1	1	0	0
B	1	0	1	0
$\bar{A}$	0	0	1	1
$\bar{B}$	0	1	0	1
$A \wedge B$	1	0	0	0
$\bar{A} \wedge \bar{B}$	0	0	0	1
$(A \wedge B) \vee (\bar{A} \wedge \bar{B})$	1	0	0	1
$A \Leftrightarrow B$	1	0	0	1

4. Igazolja igazságtábla segítségével, hogy $f(A,B,C) = (A \wedge B \wedge \bar{C}) \vee (\bar{A} \wedge B \wedge \bar{C}) \vee (\bar{A} \wedge \bar{B} \wedge \bar{C})$, ha $f(A,B,C) = „(0,1,0,0,0,1,0,1)”$.

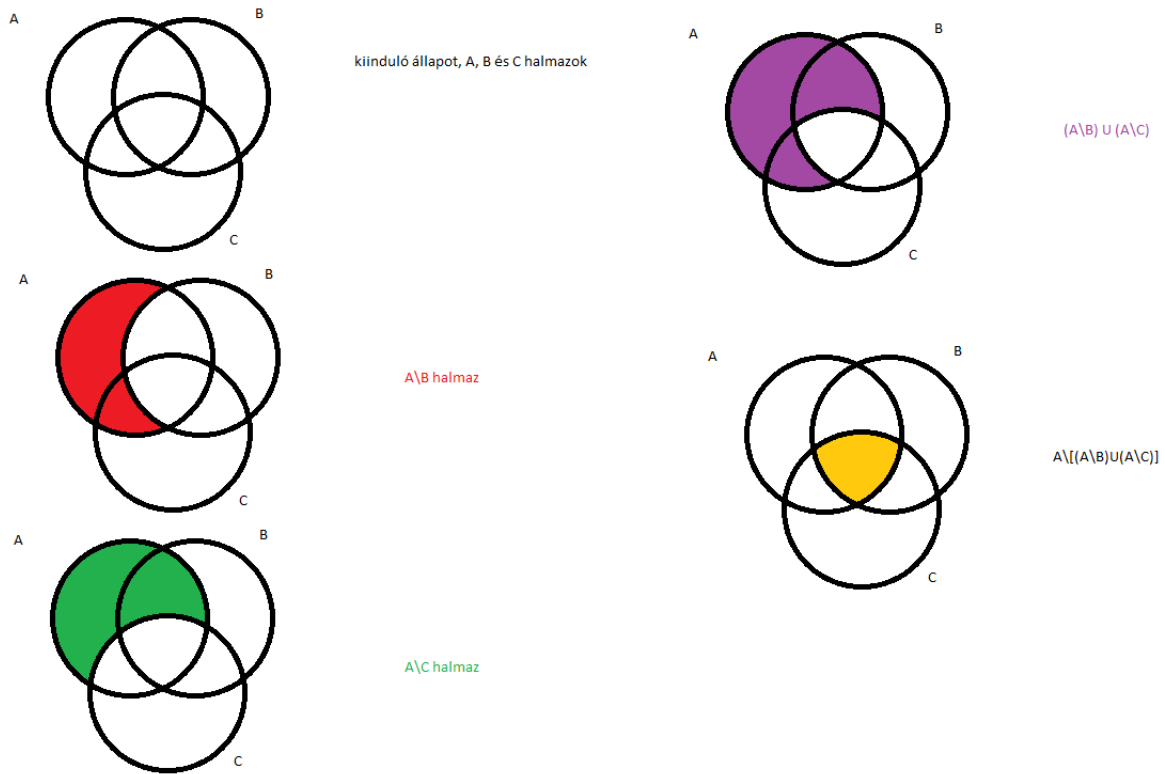
A	1	1	1	1	0	0	0	0
B	1	1	0	0	1	1	0	0
C	1	0	1	0	1	0	1	0
$\bar{A}$	0	0	0	0	1	1	1	1
$\bar{B}$	0	0	1	1	0	0	1	1
$\bar{C}$	0	1	0	1	0	1	0	1
$A \wedge B$	1	1	0	0	0	0	0	0
$A \wedge B \wedge \bar{C}$	0	1	0	0	0	0	0	0
$\bar{A} \wedge B$	0	0	0	0	1	1	0	0
$\bar{A} \wedge B \wedge \bar{C}$	0	0	0	0	0	1	0	0
$\bar{A} \wedge \bar{B}$	0	0	0	0	0	0	1	1
$\bar{A} \wedge \bar{B} \wedge \bar{C}$	0	0	0	0	0	0	0	1
$(A \wedge B \wedge \bar{C}) \vee$ $(\bar{A} \wedge B \wedge \bar{C})$	0	1	0	0	0	1	0	0
$(A \wedge B \wedge \bar{C}) \vee$ $(\bar{A} \wedge B \wedge \bar{C}) \vee$ $(\bar{A} \wedge \bar{B} \wedge \bar{C})$	0	1	0	0	0	1	0	1
f(A,B,C)	0	1	0	0	0	1	0	1

5. Hozza egyszerűbb alakra a következő halmazalgebrai kifejezést, és az eredményt ellenőrizze igazságtáblával: $(A \cap B) \cup (A \setminus B) \cup (B \setminus A)$!



Tehát $(A \cap B) \cup (A \setminus B) \cup (B \setminus A) = A \cup B$

6. Hozza egyszerűbb alakra a következő halmazalgebrai kifejezést, és az eredményt ellenőrizze igazságtáblával: $A \setminus ((A \setminus B) \cup (A \setminus C))$!



Vagyis $A \setminus ((A \setminus B) \cup (A \setminus C)) = A \cap B \cap C$

7. Legyen $A = \{1, 2, 3, 4, 5, 6\}$! Értelmezzük az $R \subseteq A \times A$ binér relációt úgy, hogy aRb akkor és csak akkor áll fenn $a, b \in A$ esetén, ha $(a-b)$ osztható 2-vel.

- R ekvivalencia reláció-e?
- R parciális rendezés-e?
- R teljes rendezés-e?
- Ábrázolja a $G=(A, R)$ gráfot!

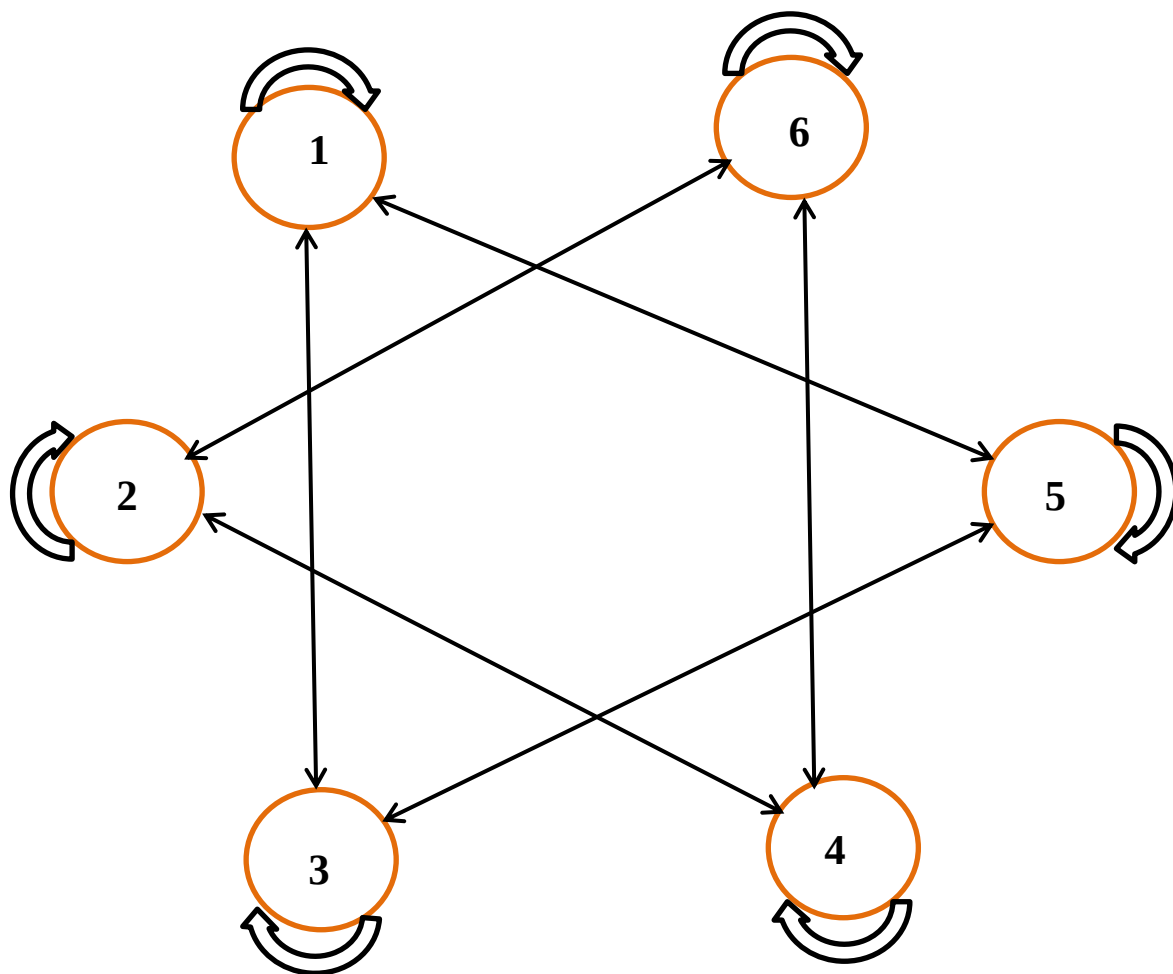
A válaszokat indokold!

Felírjuk az összes olyan párosítást, amire igaz, hogy $(a-b)$ osztható 2-vel. Ezek: $(1,1);(2,2);(3,1);(3,3);(4,2);(4,4);(5,1);(5,3);(5,5);(6,2);(6,4);(6,6);(1,3);(2,4);(1,5);(3,5);(2,6);(4,6)$.

A reláció *reflexív*, mivel aRa teljesül minden $a \in A$ elemre, vagyis ha saját magából kivonjuk a számot, akkor 0-t kapunk, a 0-nak pedig végtelen sok osztója van. A reláció *transzitiv*, mivel ha az aRb és bRc egyidejűleg teljesül, abból aRc következik, vagyis például ha $(6,4)$ és a $(4,2)$ jó, akkor a $(6,2)$ is jó. A reláció továbbá szimmetrikus, mivel a $(4,2)$ és $(2,4)$ is szerepel, de $(3,1)$ és $(1,3)$ is a felsorolásban hogy van szimmetria esetén.

Ebből következik, hogy a reláció egy ekvivalencia reláció, és nem teljes. Azért nem teljes, mert pl. $(2,5)$ de az $(5,2)$ sem tesz eleget a feltételnek.

Gráf:



8. Legyen $A = \{2,4,6,8,10,12\}$! Értelmezzük az $R \subseteq A \times A$ binér relációt úgy, hogy aRb akkor és csak akkor áll fenn $a,b \in A$ esetén, ha $(a*b)$ osztható 6-al.

A homogén bináris relációk tulajdonságai közül melyekkel rendelkezik a relációnk?

A válaszokat indokold!

Megoldás: Soroljuk fel az összes lehetséges párosítást. Ezek: $(2,6);(2,12);(4,6);(4,12);(6,2);(6,4);(6,6);(6,8);(6,10);(6,12);(8,6);(8,12);(10,6);(10,12);(12,2);(12,4);(12,6);(12,8);(12,10);(12,12)$.

Reflexivitás: Nem teljesül, mivel nem igaz minden $a \in A$ -ra, hogy ha saját magával megszorozzuk, akkor egy 6-tal osztható számot kapunk. Például $(4,4)$ nem megfelelő.

Szimmetria: Teljesül, hogy miért az tulajdonképpen a szorzás miatt van. Tehát ha veszünk a halmazból két számot (a -t és b -t), és azok szorzata osztható 6-tal, akkor nyilván ha felcseréljük őket (b és a), akkor is ugyanazt kapjuk. Tehát $(4,6)$ és $(6,4)$ itt ugyan azt jelenti, mivel a szorzatuk mindkét esetben 24, ami osztható 6-al.

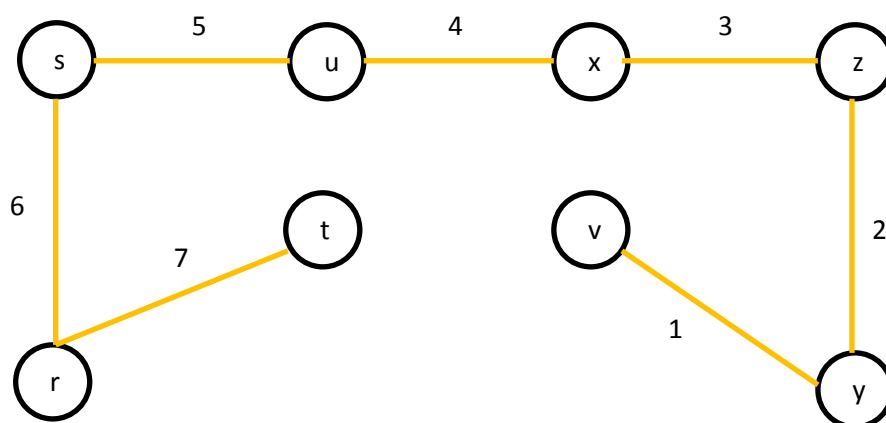
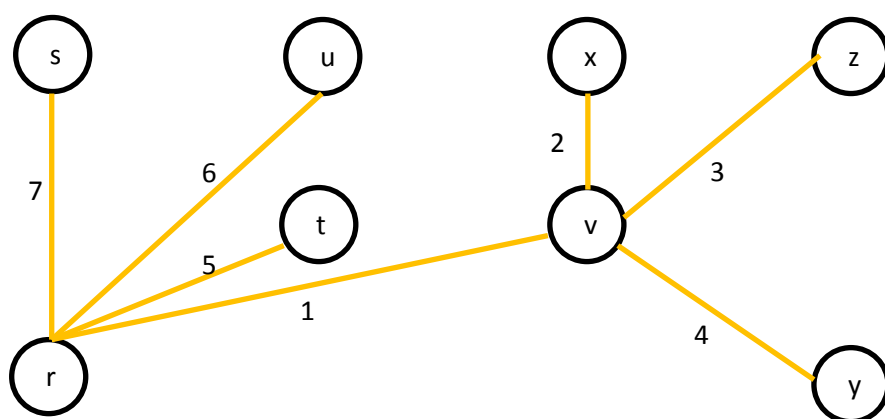
Antiszimmetria: A reláció nem antiszimmetrikus, de gondoljuk ezt mindenki kitalálta már. ☺

Tranzitivitás: A reláció ugye akkor tranzitív, ha minden $a,b,c \in A$ elemekre aRb és bRc egyidejű teljesülése esetén aRc következik. Keressünk olyan helyzetet, ahol ez nem teljesül, ha nincs, akkor esélyes hogy tranzitív. Legyen $a=4$, $b=6$, $c=8$, párosítsuk őket a feltételnek megfelelően. $aRb = 4R6$ jó, $bRc = 6R8$ szorzata szintén osztható 6-tal, viszont az $aRc = 4R8$ nem osztható 6-al, tehát a reláció nem tranzitív.

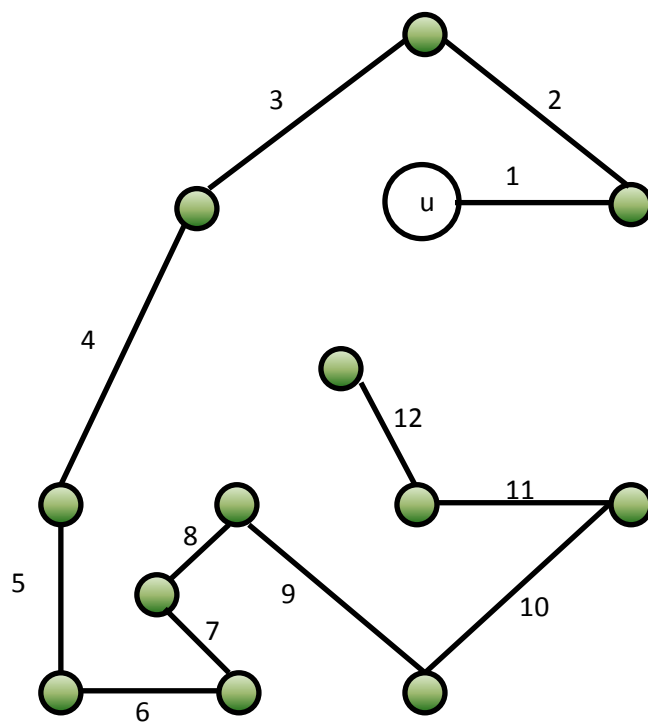
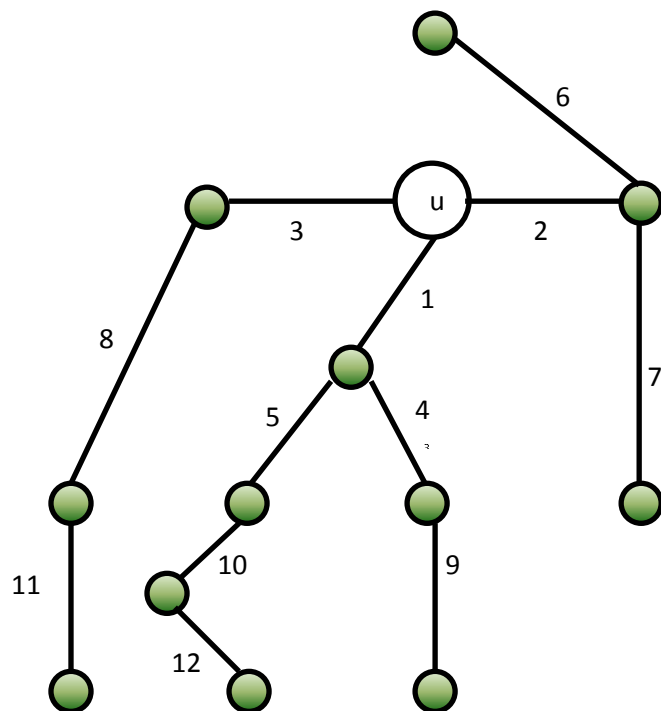
Teljesség: Nem teljes a reláció, reméljük látható, hogy miért nem az.

Gráfelmélet és Gráfalgoritmusok

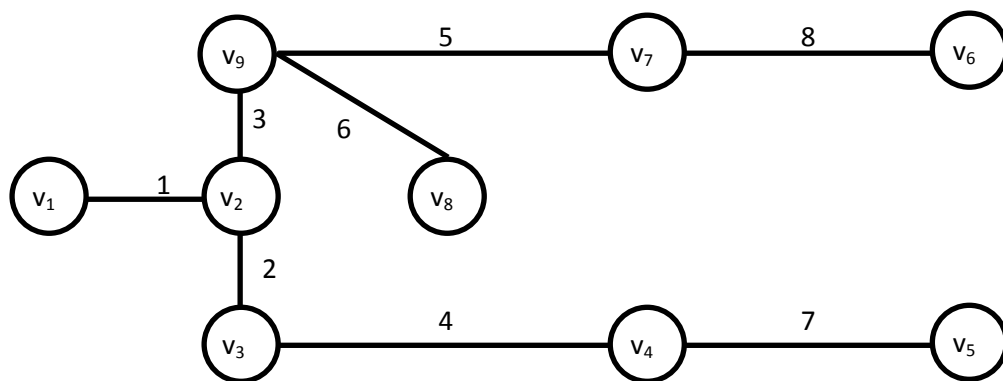
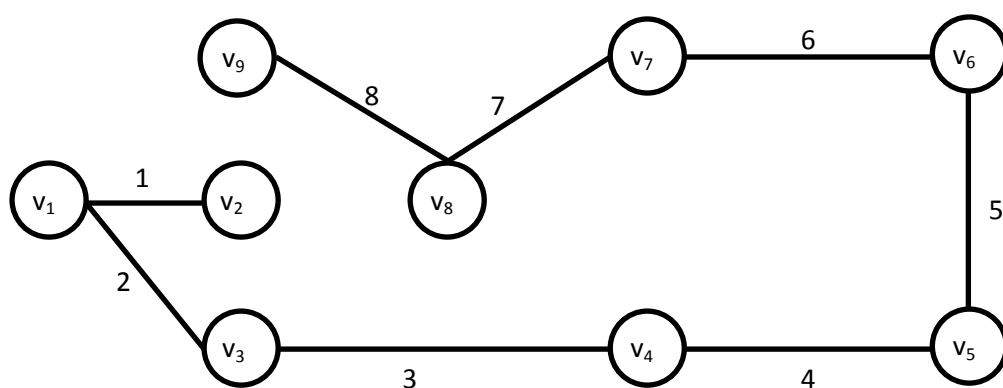
1. Szélességi, majd mélységi bejárás szerint:



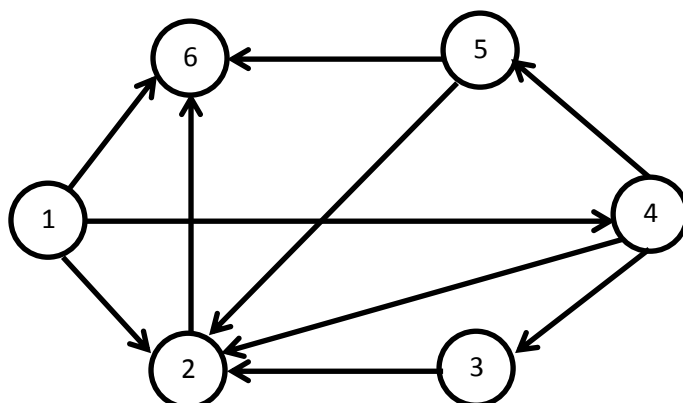
2. Előbb szélességi, majd mélységi bejárás szerint:



3. Mélységi, majd szélességi bejárás:



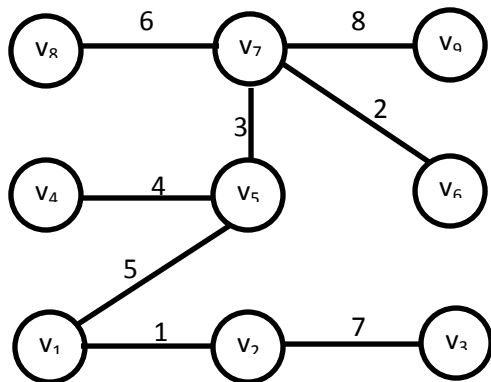
4. Irányított gráf:



Mélységi bejárás szerint a csúcsok sorrendje: 1,2,6,4,5,3.

Szélességi bejárás szerint a csúcsok sorrendje: 1,2,4,6,3,5.

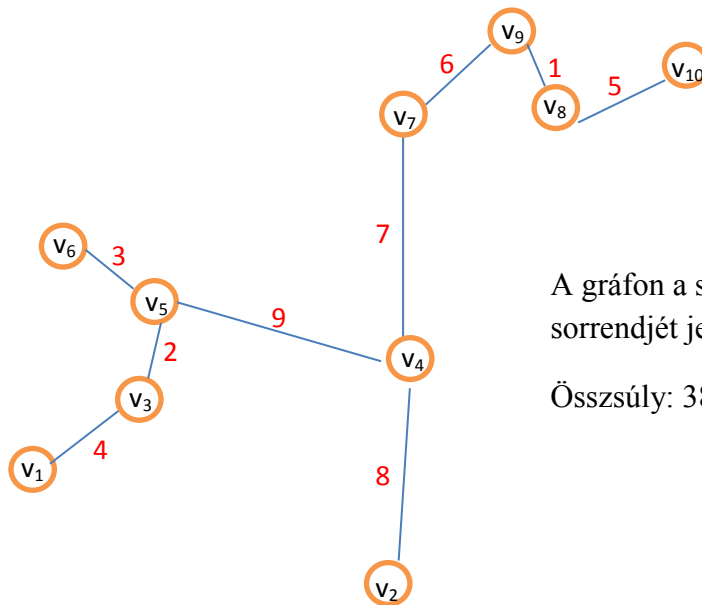
5. Minimális feszítőfa:



A gráfon a számok a kiválasztás sorrendjét jelölik.

Összsúly: 22

6. Minimális feszítőfa:



A gráfon a számok a kiválasztás sorrendjét jelölik.

Összsúly: 38

7. Legrövidebb utak

s	u	t	v	x	y	z
1,r	2,r	5,r	1,r	∞ , -	∞ , -	∞ , -
	2,r	5,r	1,r	∞ , -	5,s	∞ , -
	2,r	5,r		6,v	4,v	2, v
		5,r		6,v	4,v	2, v
		5,r		3,z	4,v	
		4,x			4,v	
					4,v	

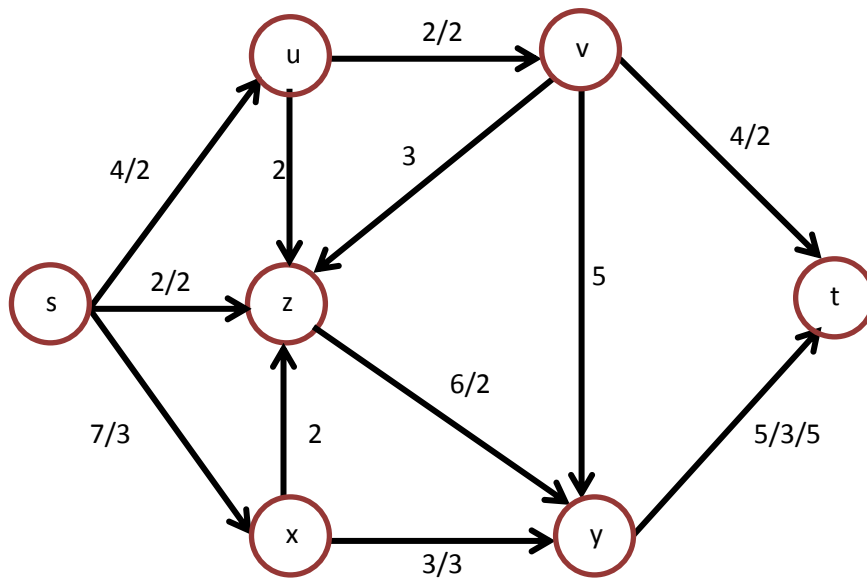
8. Legrövidebb utak: Ez egy 30 oszlopos és 29 soros táblázat lesz, akkor jó a megoldás, ha az utolsó sorban a q oszlopában $65,l$; a c oszlopában pedig $57,l$ szerepel.

9. Andi megkeresi Zolit. Ők ketten összepárosíthatók. Kati megkeresi Palit. Ők ketten is összepárosíthatók. Mari megkeresi Zolit, aki már Andival van párban, de Zoli jobban kedveli Marit Andinál, így Andi újra szabad és Mari-Zoli egy új páros. Andi ezek után Janihoz fordul, aki elfogadja őt. Szilvi megkeresi Petit, aki elutasítja őt. Ezek után Szilvi megkeresi Gyurit, aki már elfogadja őt. Tehát az Andi-Jani, Kati-Peti, Mari-Zoli és Szilvi-Zoli párosok adódnak.

A Gyuri Andit, Jani Marit és Peti Katit megkeresések sikerrel járnak. Ezek után Zoli megkeresi Marit, aki otthagyja Janit Zoli kedvéért. Most Jani Katinál próbálkozik sikertelenül. Majd Jani sikerrel jár Szilvinél. Tehát a Gyuri-Andi, Peti-Kati, Zoli-Mari és Jani-Szilvi párosok alakulnak.

Megjegyzés: két különböző stabil párosítást kaptunk. Kati-Peti és Mari-Zoli mindkét esetben párosok, de a másik két lány illetve fiú jobban járt a kezdeményező szerepkörben. (Tasnádi Attila: Számítástudomány Gazdaságinformatikusoknak, Bologna-tankönyvsorozat, 2006, 207-208. oldal)

10. Hálózati folyam:

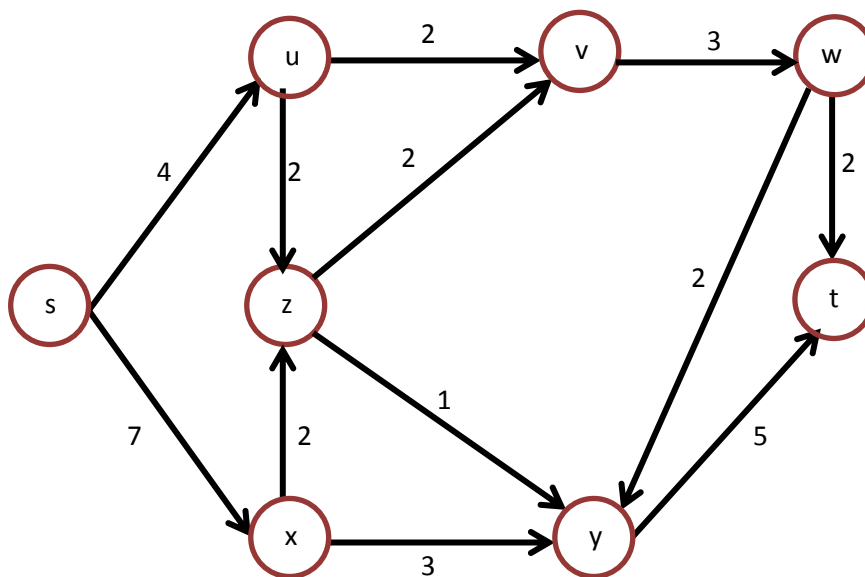


Felhasznált javító utak:

$s \rightarrow u \rightarrow v \rightarrow t$	2
$s \rightarrow x \rightarrow y \rightarrow t$	3
$s \rightarrow z \rightarrow y \rightarrow t$	2

Az $(\{s, x, y, z, u\}, \{v, t\})$ vágás értéke 7.

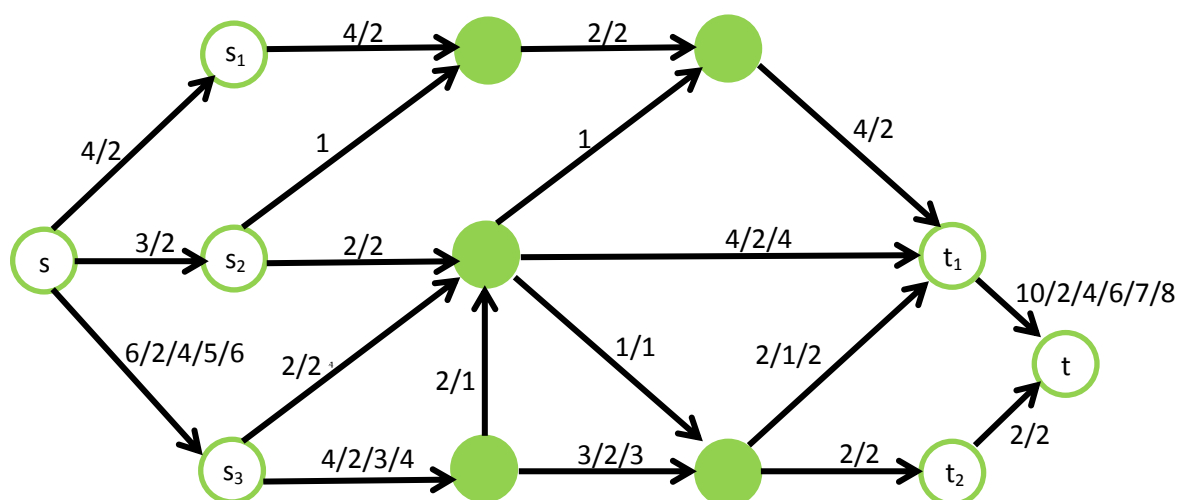
11. A csúskapacitást egy éllel helyettesítve az itt látható hálózatot kapjuk. Válasszuk a javító utakat a lent látható módon. Ezen javító utakból adódó folyam értéke 7, amely már maximális értékű, mivel a két t -be mutató él egy 7 értékű vágás.



Felhasznált javító utak:

$s \rightarrow u \rightarrow v \rightarrow w \rightarrow t$	2
$s \rightarrow x \rightarrow y \rightarrow t$	3
$s \rightarrow u \rightarrow z \rightarrow v \rightarrow w \rightarrow y \rightarrow t$	1
$s \rightarrow x \rightarrow z \rightarrow y \rightarrow t$	1

12. Felhasználtuk a feladat írója által adott ötletet, és bevezettünk egy plusz forrást, és egy plusz nyelőt. Ezt úgy oldottuk meg, hogy a folyam két „szélére” berajzoltuk az s és a t csúcsot. Az s csúcsból menő élek kapacitása 4, $1+2=3$, és $2+4=6$. Ez azért van így, mert például az eredeti ábrán az s_2 csúcsból két él indult, egy 2 és egy 1 kapacitású él. Így az általunk bevezetett s csúcsból $2+1$ értékű folyam mehet az s_2 csúcsba. A nyelő esetében pedig hasonló a helyzet, a t csúcsba $4+4+2=10$, és 2 értékű folyamok mehetnek. Ezek után a megoldás menete a szokásos módon történik, a maximális folyam pedig $2+8=10$ értékű.



Rendező algoritmusok

1. Beszúró rendezés:

6	2	1	5	7	4	2
2	6	1	5	7	4	2
1	2	6	5	7	4	2
1	2	5	6	7	4	2
1	2	5	6	7	4	2
1	2	4	5	6	7	2
1	2	2	4	5	6	7

Elég annyit csinálni, hogy minden lefutás után beírjuk a számot a megfelelő helyre. Például a 3. sorban az 1-es rögtön 2 helyet csúszik balra.

2. Összefésülő rendezés: **9,1,3,4,6,5,2,7,5.**

9,1,3,4,6 | 5,2,7,5

9,1,3 | 4,6 | | 5,2 | 7,5

9,1 | 3 | | 4,6 | | | 5 | 2 | | 7 | 5

9 | 1 | | 3 | | | 4 | 6 | | | 5 | | 2 | | | 7 | | 5

1,9 | 3 | | 4,6 | | | 5 | 2 | | 7 | 5

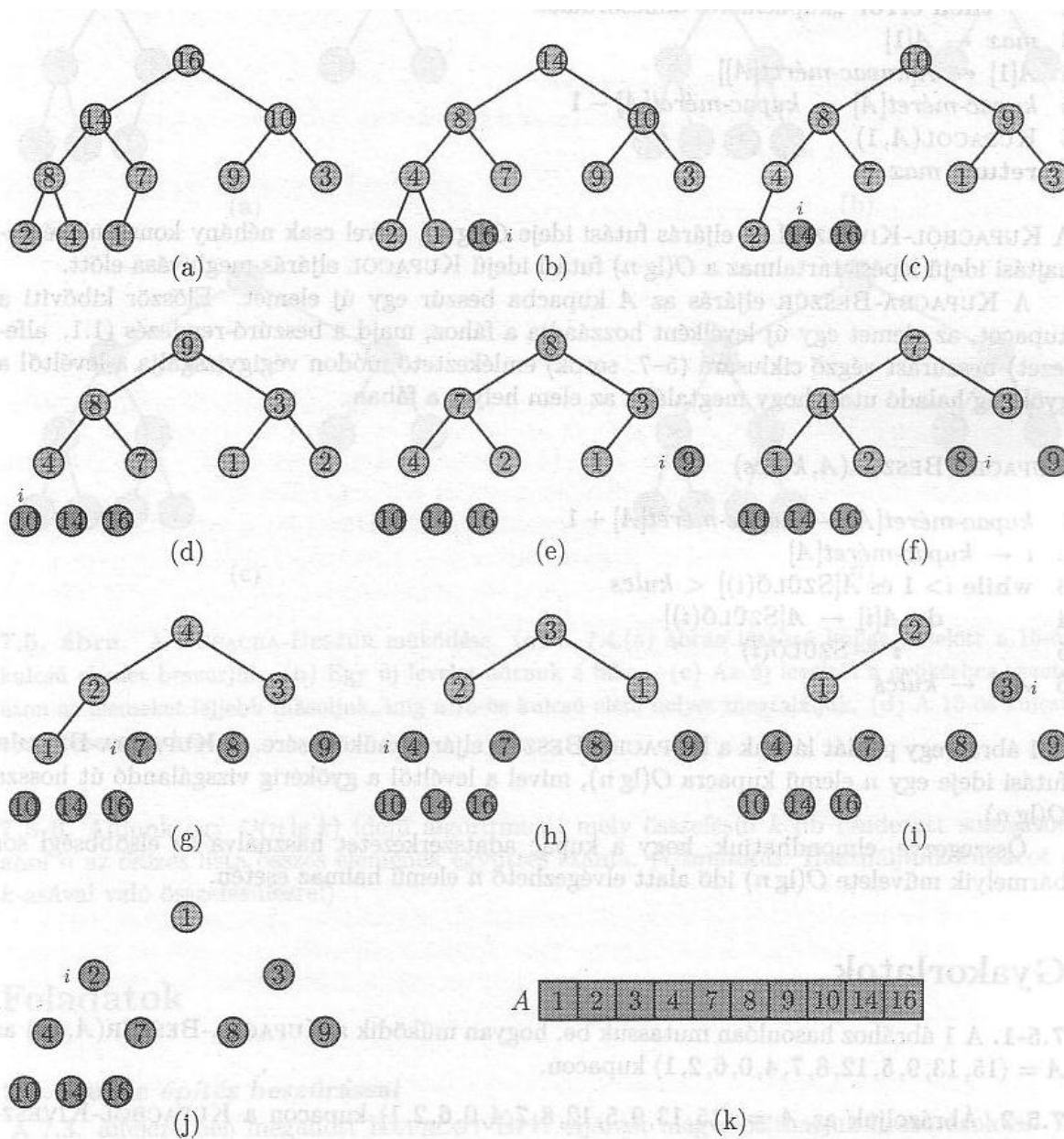
1,3,9 | 4,6 | | 2,5 | 5,7

1,3,4,6,9 | 2,5,5,7

1,2,3,4,5,5,6,7,9

3. Kupacrendezés:

Kupac: 16,14,10,8,7,9,3,2,4,1



4. Gyorsrendezés: **5,3,1,8,9,4,3**.

5	3	1	8	9	4	3
3	3	1	8	9	4	5
3	3	1	4	9	8	5
3	3	1	4	9	8	5
1	3	3	4	5	8	9
1	3	3	4	5	8	9
1	3	3	4	5	8	9
1	3	3	4	5	8	9

5. Radix rendezés:

a, Számok: **329 ; 457 ; 657 ; 839 ; 436 ; 720 ; 355**

Alap sorrend	1.	2.	3.
329	720	720	329
457	355	329	355
657	436	436	436
839	457	839	457
436	657	355	657
720	329	457	720
355	839	657	839

b, Évszámokat: 1995.06.07 ; 1996.03.12 ; 1994.03.28 ; 1993.11.07 ; 1995.05.12 ;
1997.06.28 ; 1994.10.15 ; 1995.11.12 ; 1994.12.07.

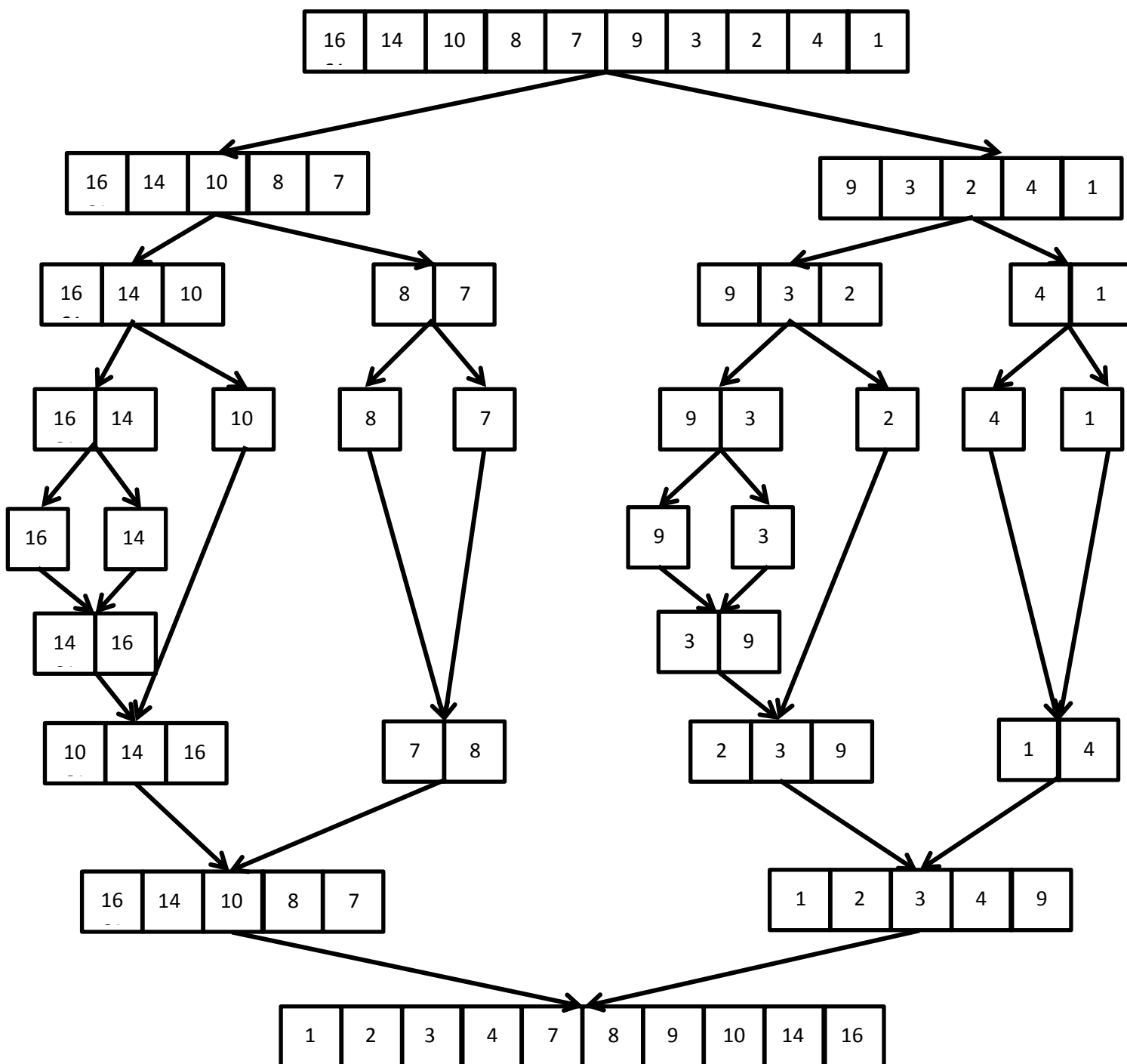
<i>Alap sorrend</i>	1.	2.	3.
1995.06.07.	1995.06.07.	1996.03.12.	1993.11.07.
1996.03.12.	1993.11.07.	1994.03.28.	1994.03.28.
1994.03.28.	1994.12.07.	1995.05.12.	1994.10.15.
1993.11.07.	1996.03.12.	1995.06.07.	1994.12.07.
1995.05.12.	1995.05.12.	1997.06.28.	1995.05.12.
1997.06.28.	1995.11.12.	1994.10.15.	1995.06.07.
1994.10.15.	1994.10.15.	1993.11.07.	1995.11.12.
1995.11.12.	1994.03.28.	1995.11.12.	1996.03.12.
1994.12.07.	1997.06.28.	1994.12.07.	1997.06.28.

6. A tömb elemei: **16,14,10,8,7,9,3,2,4,1.**

a, Beszúró rendezéssel:

16	14	10	8	7	9	3	2	4	1
14	16	10	8	7	9	3	2	4	1
10	14	16	8	7	9	3	2	4	1
8	10	14	16	7	9	3	2	4	1
7	8	10	14	16	9	3	2	4	1
7	8	9	10	14	16	3	2	4	1
3	7	8	9	10	14	16	2	4	1
2	3	7	8	9	10	14	16	4	1
2	3	4	7	8	9	10	14	16	1
1	2	3	4	7	8	9	10	14	16

b, Összefésülő rendezéssel:



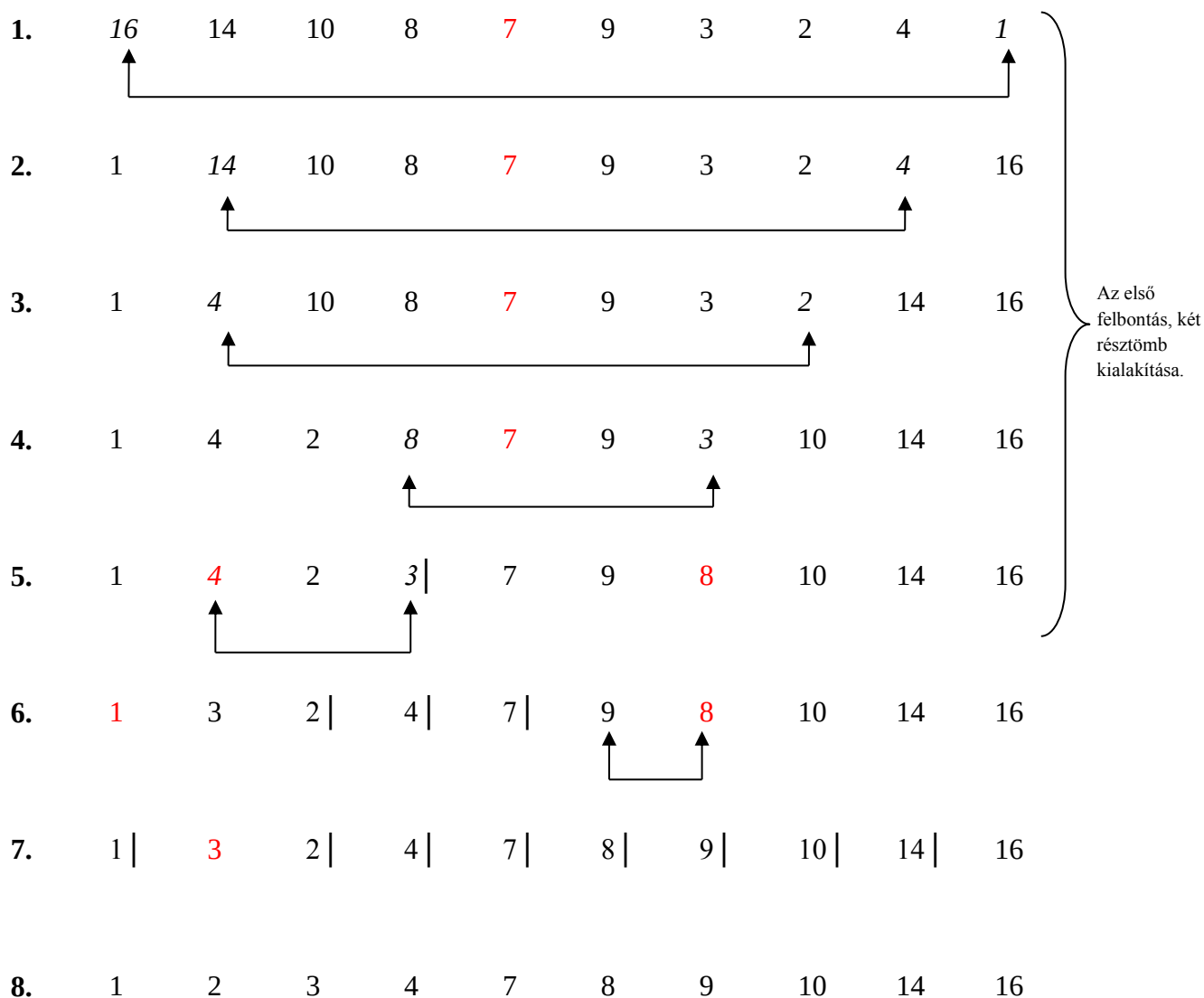
c, Gyorsrendezéssel: **16,14,10,8,7,9,3,2,4,1.**

Most oldjuk meg úgy, hogy vezérelemnek mindig a tömb középső elemét választjuk. Középső alatt azt értem, hogy a tömb elemszámát elosztjuk kettővel, és az eredményben a szám egész része által jelölt helyen lévő számot vesszük vezérelemnek (vagyis egy 10 elemű tömbben az 5. elem, 5 eleműben a 2. elem, 3 eleműben az 1. elem, 2 eleműben szintén az első elem legyen a vezérelem).

Így, a következő a megoldás:

Piros szín jelöli az aktuális tömb/tömbök vezérelemeit, dőlt betűtípus és nyíl pedig a cseréket.

A 6. sorban jön létre először a felosztás, amikor is a tömb első felében (első 4 elem) csak azok a számok vannak, amik kisebbek mint 7 (ami a legelső vezérelem), a tömb másik része pedig a maradék 6 elemet tartalmazza. A résztömbökön pedig ugyan így fut le az eljárás, amíg nem kapunk egyelem tömböket, amik már kiadják a növekvő sorrendet.



Az első vezérelem az 5. helyen álló szám, a 7-es. Hátról haladva az első nála kisebb szám az 1-es, míg előlről haladva az első nála nem kisebb szám a 16-os, így ezek helyet cserélnek.

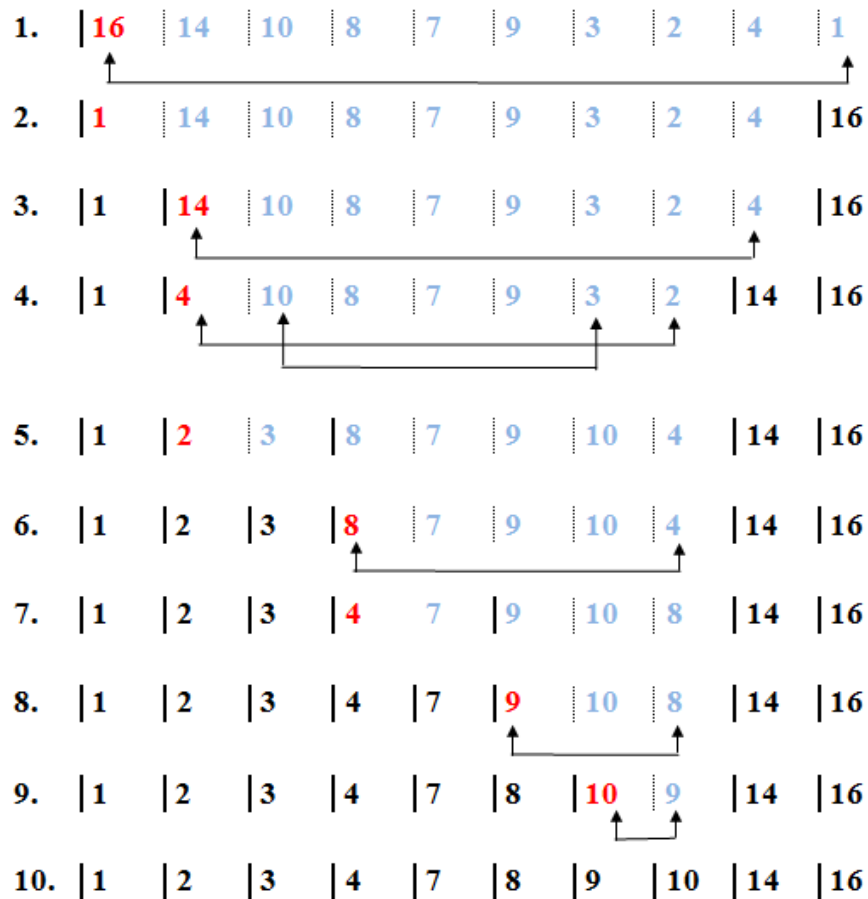
Ugyan ezen módszer alapján cserélgetnek a számok helyet egymással, és az ötödik sorban lesz először két részre bontás. A bal oldaliban a 4 darab 7-nél kisebb szám található, a jobb oldaliban pedig a 7-nél nagyobbak.

Az 5. sorban a bal oldali részre bontás vezéreleme a $(4/2=2)$ 2. helyen álló 4-es, a jobb oldali tömbben pedig a $(6/2=3)$ 3. helyen álló 8-as.

A 6. sorban látható, hogy a bal oldali tömbünk két résztömbre bomlott, az (1,3,2) és az egy elemű (4) tömbre, így a négyes véglegesen is a helyére került. Amikor az 5. sorban a 8-as volt a vezérelem, elindultunk hátulról annál kisebbet keresni, és ez a 7-es volt, ami a résztömb első eleme. Ez azt jelenti, hogy abban a tömbben a 7-es a „helyén van”, tehát szétbontható a (7,9,8,10,14,16) résztömb két további (7) és (9,8,10,14,16) tömbökre. Az (1,3,2) résztömbben pedig az 1-es lesz a vezérelem, ami az első elem, és mivel nem talál magánál kisebbet, ezért ő a legkisebb, tehát szintén a helyén van, és (1) (3,2) tömbök keletkeznek.

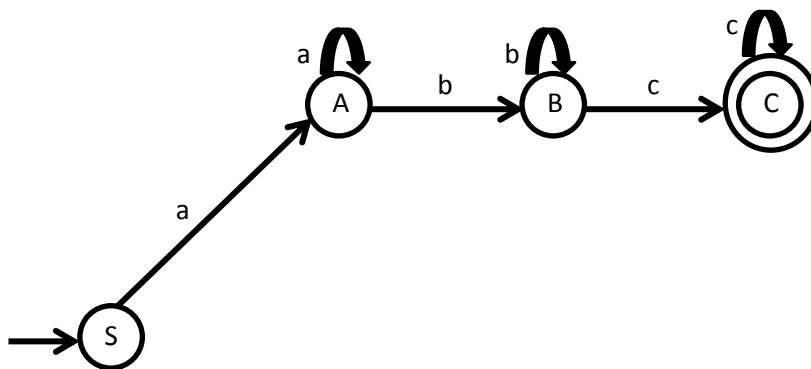
Így 8 lépésből ki tud alakulni a növekvő sorrend, ha a megoldás helyes.

A „hagyományos” módon pedig (Hornung Tamás tanár úr megoldása):



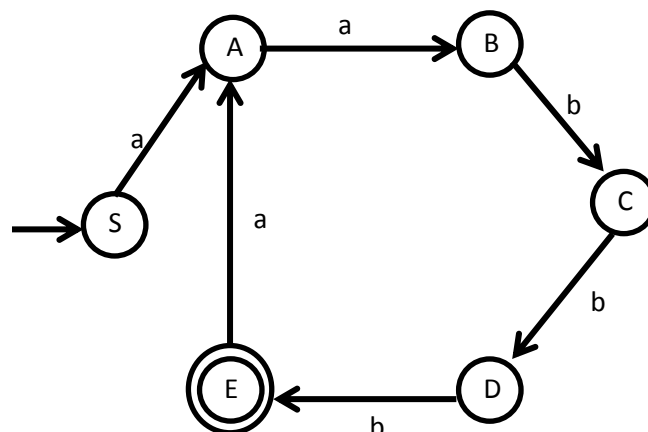
Formális nyelvek és automaták

1. Az $L = \{a^i b^i \mid i \in \mathbb{N}^+\}$ nyelvet generáló környezetfüggetlen grammatika: $G = (\{a,b\}; \{S\}; \{S\}; \{S \rightarrow aSb \mid ab\})$.
2. Nem igazán volt témája a könyvnek, de a keresett grammatika a következő: $G = (\{a,b,c\}; \{S,A,B\}; \{S\}; \{S \rightarrow aSAB \mid abB, BA \rightarrow AB, bA \rightarrow bb, bB \rightarrow bc, cB \rightarrow cc\})$.
3. A keresett nyelvtan a $G = (\{a,b\}; \{S,A,B\}; \{S\}; \{S \rightarrow aA, A \rightarrow aA \mid bB \mid b, B \rightarrow bB \mid b\})$.
4. Az $L = \{a^i b^j c^k \mid i,j,k \in \mathbb{N}^+\}$ nyelvet a $G = (\{a,b,c\}; \{S,A,B,C\}; \{S\}; \{S \rightarrow aA, A \rightarrow aA \mid bB, B \rightarrow bB \mid cC \mid c, C \rightarrow cC \mid c\})$ nyelvtan generálja.



5. ...

6. Az $L = \{(a^2 b^3)^i \mid i \in \mathbb{N}^+\}$ nyelvet generáló nyelvtan nagyon hasonló, mint ami a példafeladat volt, mint minden feladatnál itt is csak azt a logikát kell követni. Olyan grammatikát keresünk, amely az *aabbb* szavakat „rakosgatja egymás mellé”. Ilyen grammatika a $G = (\{a,b,c,d\}; \{S,A,B,C,D,E\}; \{S\}; \{S \rightarrow aA, A \rightarrow aB, B \rightarrow bC, C \rightarrow bD, D \rightarrow bE, E \rightarrow aA \mid \epsilon\})$.



7. Az 5-ös és a 7-es feladatok egymás megoldásai. Ez egy 2-es típusú grammatika.

Levezetés: $S \rightarrow aSa \rightarrow abSba \rightarrow abaaba.$

XIII. Irodalomjegyzék és ajánlott irodalom

A könyv írása során felhasznált források, irodalmak:

- Andrásfai Béla: Gráfelmélet villamosmérnökök számára, Tankönyvkiadó, Budapest, 1968, 1971.
- Andrásfai Béla: Gráfelmélet, Polygon Kiadó, Szeged, 1997.
- Andrásfai Béla: Gráfelmélet. Folyamok, mátrixok, Akadémiai Kiadó, Budapest, 1983. Angolul: Graph theory. Flows, matrices, Akadémiai Kiadó és Adam Hilger Ltd. Bristol, Philadelphia, 1991.
- Andrásfai Béla: Ismerkedés a gráfelmélettel, Tankönyvkiadó, Budapest, 1971, 1973, Angolul: Introductory graph theory, Akadémiai Kiadó, Budapest és Adam Hilger Ltd. Bristol, New York, 1977., 1984
- Cook, Stephen (1971). "The complexity of theorem proving procedures". Proceedings of the Third Annual ACM Symposium on Theory of Computing. pp. 151–158.
- Ésik Zoltán: A számítástudomány alapjai, Typotex Kiadó
- Friedl Katalin: Kiegészítő anyag az Algoritmuselmélet tárgyhoz I. 2008.május.22
- Garey, Michael R.; Johnson, David S. Computers and intractability. A guide to the theory of NP-completeness. A Series of Books in the Mathematical Sciences. W. H. Freeman and Co., San Francisco, Calif., 1979. x+338 pp. ISBN: 0-7167-1045-5
- Grötschel, Martin; Lovász, László; Schrijver, Alexander Geometric algorithms and combinatorial optimization. Algorithms and Combinatorics: Study and Research Texts, 2. Springer-Verlag, Berlin, 1988. xii+362 pp. ISBN: 3-540-13624-X
- Hajnal Péter, Gráfelmélet , II. kiadás, Polygon Jegyzet, Szeged, 2003
- Hornung Tamás Tanár Úr féle előadások a 2011-es Számítástudomány tantárgyunkhoz, BGF – GKZ:
 - Gráfalgoritmusok
 - Gráfok
 - Rendezések
- Katona – Recski – Szabó: A Számítástudomány alapja, Typotex, 5. kiadás, Budapest, 2006.
- Lovász, László An algorithmic theory of numbers, graphs and convexity. CBMS-NSF Regional Conference Series in Applied Mathematics, 50. Society for Industrial and

Applied Mathematics (SIAM), Philadelphia, PA, 1986. iv+91 pp. ISBN: 0-89871-203-3

- Lovász, László Geometric algorithms and algorithmic geometry. A plenary address presented at the International Congress of Mathematicians held in Kyoto, August 1990. ICM-90. Mathematical Society of Japan, Tokyo; distributed outside Asia by the American Mathematical Society, Providence, RI, 1990. 1 videocassette (NTSC; 1/2 inch; VHS) (60 min.); sd., col. ISBN: 0-8218-8052-7
- Tasnádi Attila: Számítástudomány Gazdaságinformatikusoknak, Bologna – Tankönyvsorozat, 2006.
- Vajk István: Formális nyelvek és automaták, oktatási segédlet, 2007.

XIV. Algoritmusok

(Az algoritmusok forrása: Hornung Tamás tanár úr féle Számítástudomány előadások, BGF-GKZ, 2011/2012-es tanév)

SZÉLESSÉGI BEJÁRÁS (G, s)

ciklus1 $\forall u \in V(G)$ -re

$Szín(u) := FEHÉR$

$D(u) := \infty$

$P(u) := NIL$

ciklus1 vége

fej := 1

vég := 1

$Q(vég) := s$

$vég := vég + 1$

$Szín(s) := SZÜRKE$

$D(s) := 0$

Ciklus2 amíg $fej = vég$

$u := Q(fej)$

ciklus3 $\forall v \in Adj(u)$ -ra

ha $Szín(v) = FEHÉR$

$q(vég) := v$

$Szín(v) := SZÜRKE$

$D(v) := D(u) + 1$

$P(v) := u$

ciklus3 vége

$fej := fej + 1$

$Szín(u) := FEKETE$

ciklus2 vége

MÉLYSÉGI BEJÁRÁS(G)

Ciklus1 $\forall u \in V(G)$ -re

$szín(u) := FEHÉR$

```

    P(u) := NIL
ciklus1 vége
ciklus2  $\forall u \in V(G)$ -re
    ha szín(u) = FEHÉR
        akkor Bejár(u)
    elágazás vége
ciklus2 vége

```

```

Bejár(u)
Szín(u) := SZÜRKE
ciklus3 /minden/ u /eleme/ Adj(u)-ra
    ha Szín(v) = FEHÉR
        akkor Bejár(v)
    elágazás vége
ciklus3 vége

```

KRUSKAL ALGORITMUSA

bemenet: G összefüggő, súlyozott gráf, $w: E(G) \rightarrow \mathbb{R}$ súlyfüggvény.

kimenet: G minimális feszítő fája.

Mff – Kruskal (G, w)

- 1 $A := \emptyset$
- 2 Ciklus $\forall u \in V(G)$ -re
- 3 HALMAZT_KÉSZÍT(u)
- 4 Ciklus1 vége
- 5 Rendezzük E éleit w szerint növekvő sorrendbe
- 6 Ciklus2 $\forall (u,v) \in E(G)$ -re a súly szerint növekvő sorrendbe
- 7 ha HALMAZT_KERES(u) $\neq$ HALMAZT_KERES(v)
- 8 akkor $A := A \cup \{(u,v)\}$
- 9 HALMAZT_EGYESÍT (u,v)
- 10 elágazás vége

11 Ciklus2 vége

DIJKSTRA ALGORITMUSA

bemenet: G gráf, w nemnegatív súlyfüggvény, s kezdőcsúcs

kimenet: $D(v) = \delta(s, v)$, $P(v)$: a v -t a legrövidebb úton megelőző csúcs (v elődje)

KEZDŐÉRTÉK(G, s)

1 Ciklus $\forall u \in V(G)$ -re

2 $D(u) := \infty$

3 $P(u) := \text{NIL}$

4 ciklus vége

5 $D(s) := 0$

KÖZELÍT(u, v, w)

1 Ha $D(v) > D(u) + w(uv)$ akkor

2 $D(v) := D(u) + w(uv)$

3 $P(v) := u$

4 elágazás vége

DIJKSTRA(G, w, s)

1 KEZDŐÉRTÉK(G, s)

2 $S := \emptyset$

3 $Q := V(G)$

4 Ciklus1 amíg $Q \neq \emptyset$

5 $u := \text{a } Q \text{ -ben, amelyre } D(u) = \min$

6 $Q := Q - \{u\}$

7 $S := S \cup \{u\}$

8 Ciklus2 $\forall v \in \text{Adj}(u)$ -ra

9 $\text{KÖZELÍT}(u, v, w)$

10 Ciklus2 vége

11 Ciklus1 vége

BESZÚRÓ_RENDEZÉS(X)

Ciklus1 $j := 2$ -től n -ig

$y := X(j)$

$i := j - 1$

Ciklus2 amíg $i > 0$ és $X(i) > y$

$X(i+1) := X(i)$

$i := i - 1$

Ciklus2 vége

$X(i) := y$

Ciklus1 vége

XV. Tanári vélemények

Természetesen elkerülhetetlen volt utánajárni, hogy mit gondolnak a tanáraink az általunk írt feladatmegoldó segédanyagról. E célból kérdeztük meg a Budapesti Gazdálkodási Főiskola – Gazdálkodási Kar Zalaegerszeg két tanárát, Dr. Szatmári Ferenc tanár urat, valamint Hornung Tamás tanár urat, aki a GKZ-n a számítástudomány tantárgyat oktatja nekünk.

A beszélgetés elején a tanár urakat kértük, hogy fejtsék ki véleményüket a Puska helyett címet viselő munkánkról.

Hornung Tamás: *A dolgozat műfaját tekintve oktatási segédlet. A 2011-ben indult gazdaságinformatikai szak Számítástudomány című tárgyához készült. Célkitűzése, hogy közérthetően, a felhasználóhoz közel álló nyelven magyarázza el a tananyagból kiválasztott témaköröket, mutassa be a kapcsolódó mintapéldák és gyakorlófeladatok megoldását. A dolgozat ezt a célkitűzést kitűnően megoldja, és mint a terjedelme is mutatja, a szerzők átfogó munkát végeztek.*

Dr. Szatmári Ferenc: *A magam részéről egészen rendkívülinek tartom ezt a feladatmegoldó példatárat, és javaslom hivatalos tananyagsegédletté beemelni a gazdaságinformatikusi oktatásba. Rendkívül jól egészítheti ki az oktatást, ahol soha nincs elegendő idő a teljes mélységű felzárkóztatásra. Nagyon komoly tárgyi tudást tükröz a munka.*

A beszélgetés további részében Hornung Tamás tanár úrral készítettünk interjút.

- Mi volt az első dolog, ami eszébe jutott, amikor tudomást szerzett a Puska helyett című dolgozat létezéséről, és témájáról?
- *Nagyon pozitívnak tartom, hogy a tárgy annyira felkeltette a szerzők érdeklődését, hogy ilyen sok munkát fektettek a tantárgy anyagának feldolgozásába. Számomra nagyon imponáló hogy vannak olyan hallgatók, akik ennyit foglalkoztak ezzel a tantárggyal.*
- Mint tudjuk, készül egy feladatmegoldó program a dolgozathoz, mert tartalmazni fog részprogramot a következő témakörökhöz: becslések, gráfalgoritmusok, rendező algoritmusok, és talán a formális nyelvek témakörhöz is. Mit gondol a készülő feladatmegoldó programról?
- *Először is, a becslések azzal függenek össze, hogy valaki érti-e azokat a fogalmakat, amiket ott bevezettünk. Ugyanis, ha meg van adva egy polinom, akkor ennek a*

nagyságendjét a legnagyobb fokú polinom határozza meg. Erre jó rálátást adhat a program, hiszen általa a felhasználó megérti akár az elméletet is, valamint programozási szempontból mindenképpen nagyon érdekes a kérdéses algoritmus. A gráfalgoritmusok és rendező eljárások pedig mindenképpen nagyon sokat segíthetnek a hallgatók felkészülésében, mivel saját készítésű feladatokat generálhatnak maguknak, melyet a program megold.

- Mennyire lehet hasznos a feladatmegoldó program, milyen mértékben segítheti az oktatást?
- *A program segíthet a hallgatóknak abban, hogy ellenőrizzék a megoldásuk helyességét. Ezáltal a Puska helyett című dolgozat használata interaktívvá válik, mert visszajelzést kapnak a megoldásukról.*
- Szívesen bevenné-e a javasolt tananyagok közé a hallgatóinak? Esetleg használná-e mint tananyag, vagy bármilyen más formában?
- *A dolgozatban tárgyalt témakörökben javasolnám a hallgatóknak a felkészüléshez, valamint oktatási segédlet formájában felhasználhatónak tartom az oktatásban.*

Ezek után megkérdeztük a beszélgetőpartnerünket arról, hogy mit gondol a dolgozat jövőjéről, valamint mi az, amit javasolna ezzel kapcsolatban, milyen fejlesztéseket ajánlana? A tanár úr elmondta, hogy szerinte a jövőben érdemes lenne bővíteni a dolgozat által tárgyalt témaköröket, és az ezekhez kapcsolódó feladatok listáját. Jó ötletnek tartaná, ha az anyagot a hallgatók folyamatosan fejleszthetnék, és saját maguk hozhatnának létre egy tudásbázist.

Elmondása alapján megalapozottnak látszik, hogy a munkát a terveink szerint érdemes folytatni. Bízunk benne, hogy egyre több hallgató bekapcsolódik majd ebbe, ez felkelti az érdeklődést a tárgy iránt és hogy sokat tudják sikeresen alkalmazni majd a vizsgákra való felkészülés során is.

XVI. Az általunk tanult Számítástudomány tantárgyi tematikája, BGF-GKZ

A tantárgy képzési célja:

„ Megismertetni a hallgatókkal a számítási folyamatok matematikai vonatkozásait, áttekinteni az informatikában alkalmazott legfontosabb matematikai eszközöket.

Bemutatni az algoritmuselmélet és a gráfelméleti alapjait. Bevezetni a hallgatókat az automaták és formális nyelvek elméletébe. Ismertetni a kiszámíthatóság és bonyolultság elméletének elemeit.”

A tananyag tartalma részletesen:	
I. Algoritmusok. Gráfok	
1.	Bevezetés Egyszerű algoritmusok. Műveletigény aszimptotikus vizsgálata.
2.	Logikai, halmazelmélet Műveletek. Azonosságok. Igazságtábla. Diszjunktív normál forma. Relációk. Ekvivalencia reláció, parciális rendezés. Függvények. Számosságok.
3.	Gráfok Fogalmak. Síkbarajzolhatóság. Színezés. Párosítás.
4.	Gráfalgoritmusok 1. Szélességi és mélységi bejárás. Minimális feszítőfa (Kruskal algoritmusa).
5.	Gráfalgoritmusok 2. Legrövidebb utak (Dijkstra algoritmusa). Maximális folyam (Ford és Fulkerson módszere). Kritikus út.
6.	Rendezések Beszűrő, összefésülő, kupac-, gyorsrendezés. Összehasonlító rendezések. Rendezés

	lineáris időben.
7.	Adatszerkezetek Lineáris adatszerkezetek, vermek és sorok. Hasítás. Keresőfák. Gráfok ábrázolása.
II. Automaták és formális nyelvek	
8.	Formális nyelvek Nyelvtanok. Chomsky-féle nyelvosztályok. Reguláris nyelvek.
9.	Véges automaták Nyelvfelismerés véges automatákkal. Mealy- és Moore-automaták.
10.	Környezetfüggetlen nyelvek Tulajdonságok. Veremautomaták.
11.	Turing-gépek. Nyelvfelismerés Turing-géppel. Rekurzív és rekurzívan felsorolható nyelvek.
12.	Kiszámíthatóság és bonyolultság. Turing-gépek, mint a számítás formális modelljei. Church tézise. Eldönthetetlen problémák. Algoritmusok idő- és tárigénye. Bonyolultsági osztályok. A P és NP osztályok. P-teljes és NP-teljes problémák.

(2012/2013-as tanév, BGF-GKZ, Számítástudomány kurzus, oktató: Hornung Tamás)