Bonyolultságelmélet gyakorlat – 10
Approximálás I.

Recap: Approximálás

Egy **optimalizálási** problémánál minden I inputhoz tartozik **megoldások** egy $S(I)$ halmaza, és minden $s \in S(I)$ megoldásnak adott a $c(s)$ **költsége** vagy **haszna**, attól függően, hogy minimalizálási vagy maximalizálási problémáról van szó.

Egy **approximáló algoitmus**tól a következőket várjuk el:

- fusson polinomidőben,
- minden I inputra egy $s \in S(I)$ tényleges megoldást adjon vissza,
- úgy, hogy a visszaadott megoldás költsége/haszna garantáltan az optimumhoz képest egy szorzón belül maradjon.

Konkrétan, ha $\alpha \geq 1$ egy konstans, akkor az A algoritmus α-*approximáló*, ha minden I esetén

$$c(A(I)) \in \left[\frac{1}{\alpha} \cdot \mathrm{opt}(I),\ \alpha \cdot \mathrm{opt}(I)\right],$$

ahol $\mathrm{opt}(I)$ a feladat optimális megoldásának a költsége/haszna.

Minimalizálási problémánál persze (mivel az optimálisnál jobb megoldás nem létezik) ez azt jelenti, hogy a visszaadott megoldás költsége legfeljebb α-szor annyi, mint az optimum; maximalizálásinál pedig azt, hogy a haszna legalább az optimum α-ad része lesz.

Recap: Hátizsák

A Hátizsák probléma optimalizálási változata:

- **Input:** N darab tárgy súlya és értéke (w_i, c_i) (egész) számok sorozataként, és egy $W \geq 0$ (egész) összkapacitás.
- **Megoldás:** a tárgyaknak tetszőleges olyan $S \subseteq \{1, \ldots, N\}$ részhalmaza, melyek összsúlya a kapacitáskorláton belül marad, azaz melyre $\sum_{i \in S} w_i \leq W$.
- **A megoldás értéke:** az S megoldásbeli tárgyak össz értéke, azaz $\sum_{i \in S} c_i$.

A probléma természetesen egy **maximalizálási** feladat.

Recap: Mohó algoritmus

- Rendezzük a tárgyakat **csökkenő sorrendbe** a $\frac{c_i}{w_i}$ **fajlagos értékük** szerint
- Ebben a sorrendben próbáljuk meg berakni őket a hátizsákba: először a fajlagosan legdrágábbat, majd a másodikat stb; aki nem fér be, kihagyjuk és folytatjuk a többivel sorrendben.

Ez az algoritmus magában még **nem** α-approximálja a problémát semmilyen α konstansra, de egy kis módosítással már igen, sőt α tetszőlegesen közel vihető lesz 1-hez:

Recap: $\left(1+\frac{1}{k}\right)$-approximáló algoritmus a Hátizsákra

Itt a $k \geq 1$ egész szám egy paraméter konstans, melyet mi választunk meg; minél jobban akarjuk approximálni az optimumot, annál nagyobbra kell vegyük (és annál nagyobb, de minden k esetén még polinom) lesz a futásidő.

- Ahányféleképp csak lehet, válasszunk ki *legfeljebb* k tárgyat, legyen a kiválasztott halmaz S.
- Ha a kiválasztott tárgyak egyszerre beférnek (azaz ha $\sum_{i\in S} w_i \leq W$), akkor a **többi** tárgyat a mohó algoritmus szerint válogassuk ki a hátizsák megmaradt kapacitásába.
- A kapott megoldások közül adjuk vissza a legjobbat.

Ez pl. $k=1$ esetén tehát azt jelenti, hogy mindegyik (w_i, c_i) tárgyra megnézzük, hogy ha ezt a tárgyat mindenképp elvisszük, a többit pedig mohón pakoljuk, akkor mit kaphatunk legjobb esetben: a fenti szerint ez egy $\left(1+\frac{1}{1}\right)$, azaz 2-approximáló algoritmus lesz.

Pár megjegyzés:

- Érdemes a tárgyakat csak **egyszer** lerendezni és a rendezett sorrendet megtartani
- Ha egy k-elemű S részhalmaz egyben befér, akkor ennek nem kell megnézzük a valódi részhalmazait
- Ha egy tárgy magában nem fér be a hátizsákba, akkor őt törölhetjük is a tárgyak listájáról

1. feladat.

Futtasuk a fenti algoritmust olyan paraméterezéssel a következő inputon, hogy 2-approximálást kapjunk! Mit tudunk mondani az optimum értékéről az eredmény alapján?

tárgy	A	B	C	D
súly	100	10	50	50
érték	150	20	55	50

és $W = 100$.

2. feladat.

Futtasuk a fenti algoritmust olyan paraméterezéssel a következő inputon, hogy 1.5-approximálást kapjunk! Mit tudunk mondani az optimum értékéről az eredmény alapján?

tárgy	A	B	C	D	E
súly	1	3	2	5	4
érték	1	2	5	4	5

és $W = 9$.

Recap: Súlyozott Csúcslefedés

- **Input:** egy $G = (V, E)$ (irányítatlan) gráf, és minden $v \in V$ csúcsának egy $c(v) > 0$ (egész) ára.
- **Megoldás:** a G gráfnak egy $S \subseteq V$ lefogó csúcshalmaza.
- **Egy megoldás költsége:** a kiválasztott csúcsok össz ára, vagyis ha S-t választjuk ki, akkor $\sum_{v \in S} c(v)$.

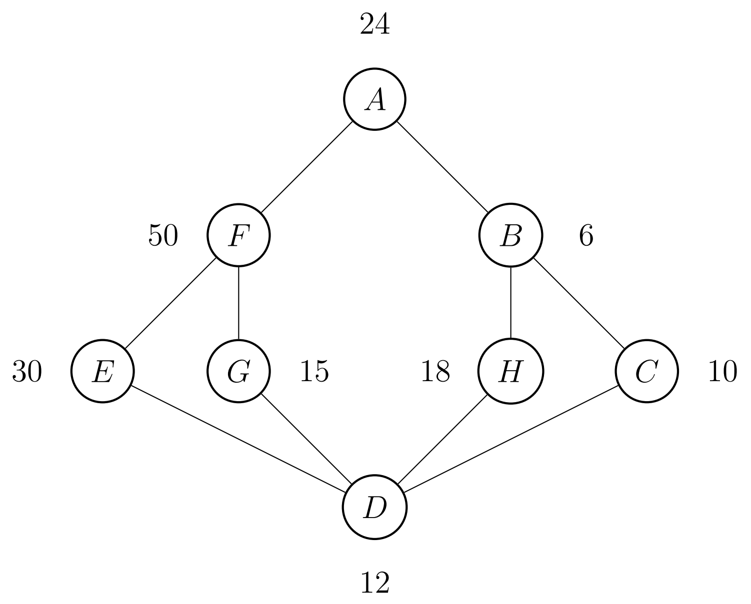
Természetesen ez is egy minimalizálási feladat (melynek eldöntési változata is **NP**-nehéz, így van értelme approximálni.)

Recap: 2-approximáló algoritmus a Súlyozott Csúcslefedésre

- Legyen $S = \emptyset$. (Ebbe a halmazba fogjuk gyűjteni a kiválasztott csúcsokat.)
- Amíg van él G-ben:
  - Válasszunk egy (u, v) élt G-ben (mindegy, melyiket). Legyen $c(u) \leq c(v)$.
  - Ha $c(u) = c(v)$, akkor vegyük be u-t is **és** v-t is S-be.
  - Ha $c(u) < c(v)$, akkor vegyük be u-t S-be és **csökkentsük** $c(v)$-t $c(u)$-val.
  - Töröljük G-ből az újonnan bevett csúcs(ok)ra illeszkedő összes élt.
- Ha elfogytak az élek, adjuk vissza S-t.

3. feladat.

Futtassuk a 2-approximáló algoritmust a Súlyozott Csúcslefedés következő példányán, ahol a csúcsok értékét a melléjük írt számok jelzik:



Mit tudunk mondani az optimumról a megoldásunk ismeretében?

1. feladat megoldása.

Fajlagos érték szerint rendezve B a legdrágább (a hányados ott 2), majd A következik (1.5), ezután C (1.1), végül D (1), ebben a B, A, C, D sorrendben fogjuk a nem-kiválasztott tárgyakat megpróbálni berakni a hátizsákba.

Mivel 2-approximálást keresünk, így $k = 1$-gyel elég futtatni az algoritmust, azaz az **egyelemű** halmazokat rögzítjük le (azt mindegy, milyen sorrendben, most ábécében fogjuk):

- Ha A-t választjuk be, akkor a hátizsákunkban nem marad helyünk (mivel $w_A = 100$), persze végig próbálhatjuk (ebben a sorrendben), hogy B se fér be, C se és D se, és kapjuk, hogy ekkor a megoldás $\{A\}$ lesz, melynek haszna 150.
- Ha B-t választjuk be, akkor a hátizsákunkban még 90 hely marad. Ezek után a többi tárgyat A, C, D sorrendben megpróbálva berakni:
  - A nem fér be, kihagyjuk;
  - C befér, berakjuk, marad $90 - 50 = 40$ helyünk;
  - D már nem fér be, kihagyjuk.

  Ekkor a megoldásunk $\{B, C\}$ lesz, értéke $20 + 55 = 75$ lesz. (Note: ez pont a mohó algoritmus lenne rögzítés nélkül, láthatjuk, hogy mennyivel rosszabb, mint az előző esetben amit kaptunk.)
- Ha C-t választjuk be, akkor a hátizsákunkban még 50 hely marad. Ezek után a többi tárgyat B, A, D sorrendben megpróbálva berakni:
  - B befér, berakjuk, marad $50 - 10 = 40$ helyünk;
  - ezután sem A, sem D nem fér már be.

  Ekkor a megoldásunk megint $\{B, C\}$ lesz, szintén 75-ös értékkel.
- Végül ha D-t választjuk be, akkor a hátizsákunkban még 50 hely marad és a többi tárgyat B, A, C sorrendben megpróbálva berakni:
  - B befér, berakjuk, marad 40 helyünk.
  - Ezután sem A, sem C már nem fér be.

  Ekkor a megoldásunk $\{B, D\}$ lesz, $20 + 50 = 70$-es értékkel.

A négy eset közül a legjobb az $\{A\}$, melynek az értéke 150. Abból, hogy az algoritmus 2-approximáló, annyit tudunk megállapítani biztosan, hogy az optimum értéke legfeljebb $2 \cdot 150 = 300$ (persze mivel a tárgyak össz értéke csak 275, ez most egy jobb korlátot ad; a mohó algoritmus lefuttatása és az utolsó tárgy eltörése pedig ad egy $20 + 150 \cdot 0.9 = 155$-ös korlátot, így most azt tudjuk ennyiből biztosan, hogy az optimum értéke 150 és 155 közé esik.)

2. feladat megoldása.

Most mivel 1.5-approximálás a feladat, és $1 + \frac{1}{2} \leq 1.5$, így $k = 2$ paraméterezéssel futtatjuk az algoritmust ($k = 1$ még csak 2-approximálást tudna garantálni).

Rendezve a tárgyakat fajlagos érték szerint csökkenőbe a következő sorrendet kapjuk: C, E, A, D, B.

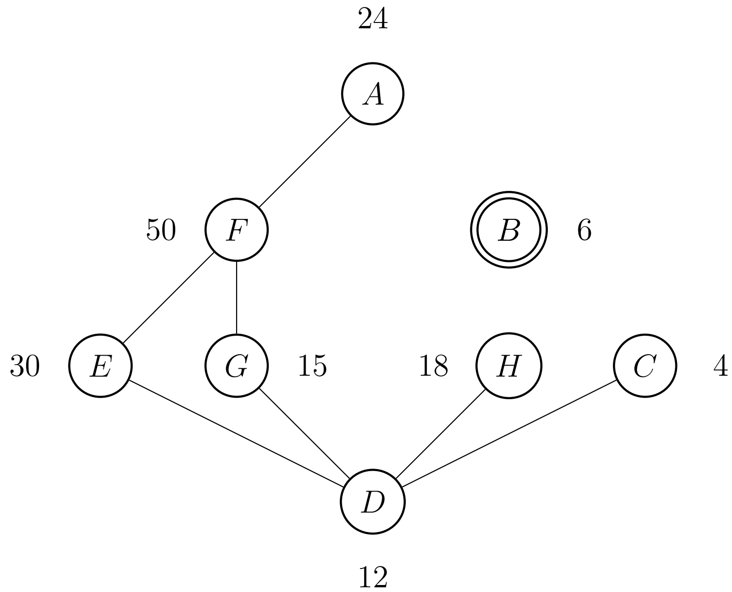
- Ha az $\{A, B\}$ tárgyakat rakjuk be, akkor marad 5 helyünk, ide C, E, D sorrendben megpróbálva berakni a tárgyakat C befér (marad 3 hely), E és D már nem fér be. Ekkor a megoldásunk $\{A, B, C\}$ lesz, melynek haszna $1 + 2 + 5 = 8$.
- Ha az $\{A, C\}$ tárgyakat rakjuk be, akkor marad 6 helyünk, ide E, D, B sorrendben megpróbálva berakni a tárgyakat E befér (marad 2 hely), D nem fér be, B sem. Ekkor a megoldásunk $\{A, C, E\}$ lesz, melynek haszna $1 + 5 + 5 = 11$.
- Ha az $\{A, D\}$ tárgyakat rakjuk be, akkor marad 3 helyünk, ide C, E, B sorrendben megpróbálva berakni a tárgyakat C befér (marad 1 hely), E és B már nem. Ekkor a megoldásunk $\{A, C, D\}$ lesz, melynek haszna $1 + 5 + 4 = 10$.
- Ha az $\{A, E\}$ tárgyakat rakjuk be, akkor marad 4 helyünk, ide C, D, B sorrendben megpróbálva berakni a tárgyakat C befér (marad 2 hely), D és B már nem. Ekkor a megoldásunk $\{A, C, E\}$ lesz, ez már volt, haszna 11.
- Ha a $\{B, C\}$ tárgyakat rakjuk be, akkor marad 4 helyünk, ide E, A, D sorrendben megpróbálva berakni a tárgyakat E befér (marad 0 hely), A és D már nem. Ekkor a megoldásunk $\{B, C, E\}$ lesz, melynek haszna $2 + 5 + 5 = 12$.
- Ha a $\{B, D\}$ tárgyakat rakjuk be, akkor marad 1 helyünk, ide C, E, A sorrendben megpróbálva berakni a tárgyakat C nem fér be, E sem, de A igen (marad 0 hely). Ekkor a megoldásunk $\{A, B, D\}$ lesz, melynek haszna $1 + 2 + 4 = 7$.
- Ha a $\{B, E\}$ tárgyakat rakjuk be, akkor marad 2 helyünk, ide C, A, D sorrendben megpróbálva berakni a tárgyakat C befér (marad 0 hely), A és D már nem. Ekkor a megoldásunk $\{B, C, E\}$ lesz, ez már volt, haszna 12.
- Ha a $\{C, D\}$ tárgyakat rakjuk be, akkor marad 2 helyünk, ide E, A, B sorrendben megpróbálva berakni a tárgyakat E nem fér be, A befér (marad 1 hely), B már nem fér be. Ekkor a megoldásunk $\{A, C, D\}$ lesz, ez már volt, haszna 10.
- Ha a $\{C, E\}$ tárgyakat rakjuk be, akkor marad 3 helyünk, ide A, D, B sorrendben megpróbálva berakni a tárgyakat A befér (marad 2 hely), D és B már nem. Ekkor a megoldásunk $\{A, C, E\}$ lesz, ez már volt, haszna 11. (Ez a 0 tárgyat rögzítő mohó pakolásnak megfelelő megoldás egyébként.)
- Végül, ha a $\{D, E\}$ tárgyakat rakjuk be, akkor nem marad helyünk, a többiek nem fognak beférni. Ekkor a megoldásunk $\{D, E\}$ lesz, melynek haszna $4 + 5 = 9$.

A lehetőségek közül a legjobb a $\{B, C, E\}$ halmaz, melynek haszna 12, ezt adjuk vissza. Mivel 1.5-approximálunk, ebből azt tudjuk megállapítani, hogy az optimum értéke legfeljebb $1.5 \cdot 12 =$

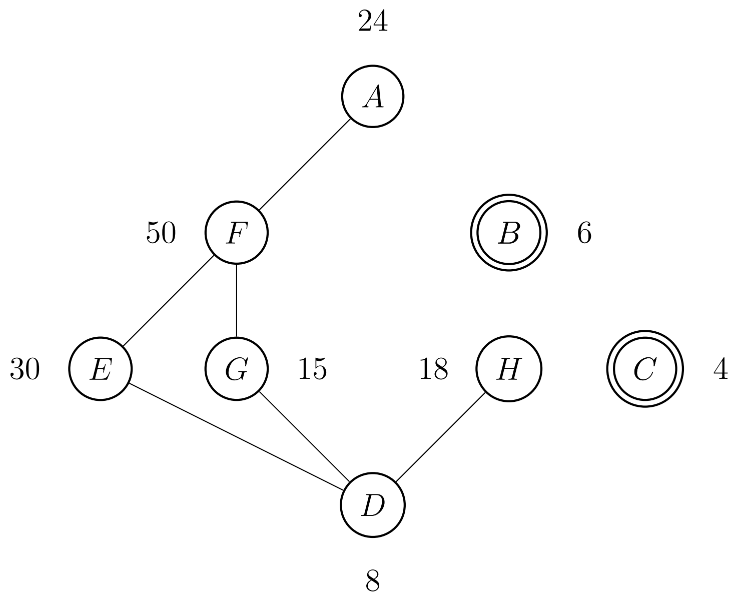
18. A mohó algoritmusnak azzal a futtatásával, melyben az utolsó tárgyat eltörjük, azt kapjuk, hogy annak a haszna (ami szintén felülről korlátozza az optimumot) $5 + 5 + 1 + 4 \cdot 2/5 = 12.6$ (mert a C, E, A tárgyak teljesen beférnek, D-nek pedig 2 hely marad és 5 a mérete, így az értékének 2/5-ét vinnénk el a töredékes változatban). Ha pedig azt tudjuk, hogy az optimum értéke legfeljebb 12.6, akkor azt is tudjuk, hogy legfeljebb 12, hiszen minden tárgy értéke egy egész szám. Azaz, most az approximáló algoritmusunk ténylegesen egy optimális megoldást adott vissza.

3. feladat megoldása.

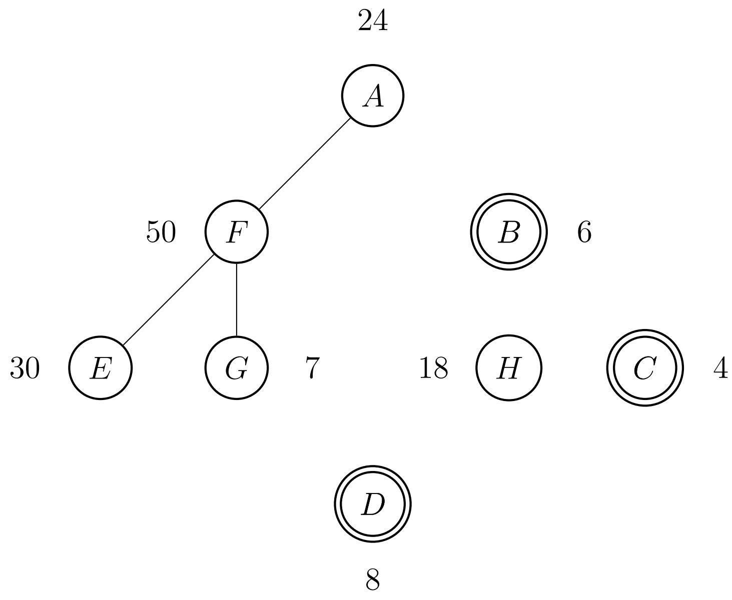
Az első lépésben ki kell választanunk egy élt, válasszuk mondjuk a (B, C) élt. Itt a két csúcs ára nem egyenlő, B az olcsóbb. Kiválasztjuk tehát B-t (ezt most dupla karikával jelöljük), C árát csökkentjük B árával (azaz 6-tal), és töröljük a B-re illeszkedő éleket:



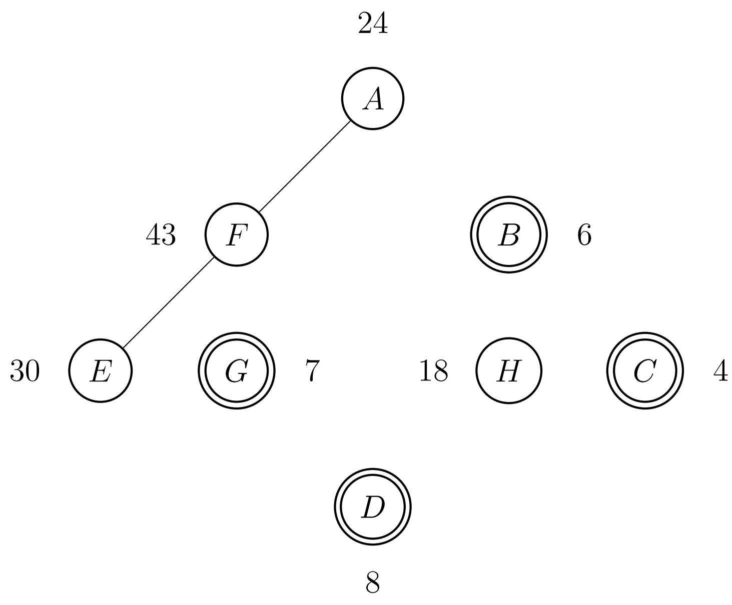
Követhetjük pl. azt a heurisztikát, hogy egy olyan élt választunk, melynek az egyik végpontja a legolcsóbb csúcs, másik pedig ennek a csúcsnak a legolcsóbb szomszédja. Ekkor most a (C, D) élt választjuk ki, C ténylegesen olcsóbb, az ő árával (azaz most 4-gyel) csökkentjük D árát virtuálisan, és töröljük a C-re illeszkedő éleket:



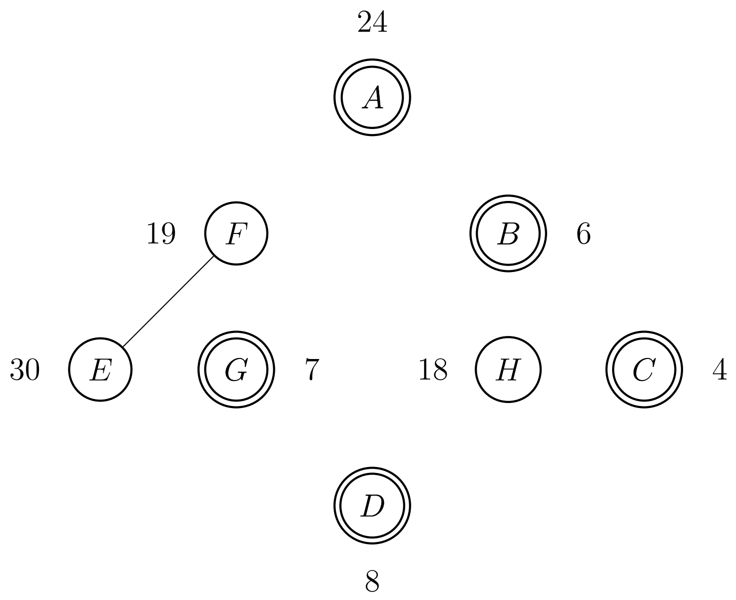
Követve ezt a stratégiát, most a (D, G) élt választjuk, D olcsóbb, mint G, ezért őt választjuk ki, G virtuális árát pedig csökkentjük D virtuális árával, azaz 8-cal, és töröljük a D-re illeszkedő éleket:



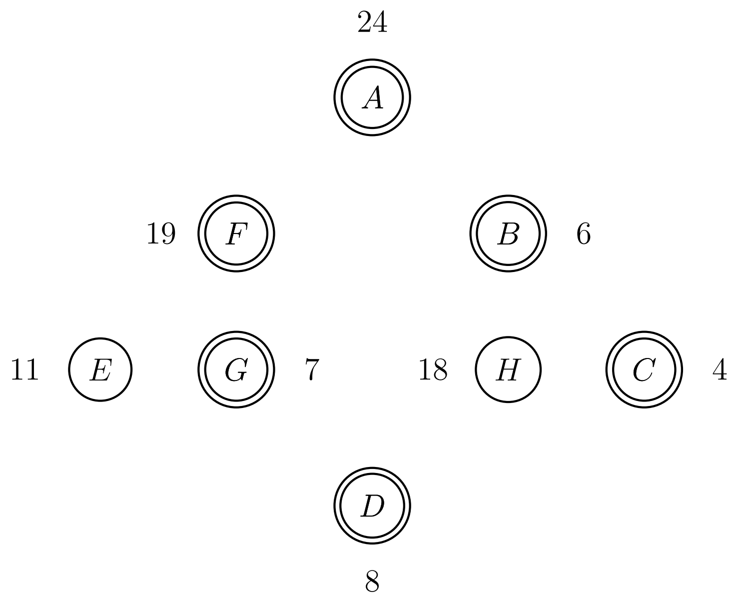
Most a G csúcs a legolcsóbb, az egyetlen rá illeszkedő él a (G, F), F határozottan drágább, mint G, így csak G-t választjuk ki, töröljük a rá illeszkedő éleket és F árát csökkentjük 7-tel:



Ezúttal az A csúcs a legolcsóbb, választjuk az (A, F) élt, A olcsóbb, mint F, ezért csak A-t választjuk ki, csökkentjük F árát 24-gyel, töröljük az A-ra illeszkedő élt:



Végül, az egyetlen megmaradt (E, F) él esetében az F csúcsot választjuk ki:



A visszaadott megolásunk az $\{A, B, C, D, F, G\}$ lefogó csúcshalmaz, melynek költsége (itt már az eredetit kell nézni!) $24 + 6 + 10 + 12 + 50 + 15 = 117$. Ebből azt tudjuk levezetni, mivel ez egy 2-approximáló algoritmus, hogy az optimum költsége legalább $\frac{1}{2} \cdot 117 = 58.5$ lesz.

Megjegyzés.

Hasonlóan a töredékes hátizsákhoz, erre is felírhatunk egy LP feladatot, melyet megoldhatunk a valós számok fölött (azaz relaxálhatjuk a feladatot), és ez is adni fog nekünk egy alsó korlátot az optimum értékére. Az LP feladat most:

$$
\begin{aligned}
x_A + x_F &\geq 1 \\
x_A + x_B &\geq 1 \\
x_B + x_C &\geq 1 \\
x_B + x_H &\geq 1 \\
x_C + x_D &\geq 1 \\
x_D + x_E &\geq 1 \\
x_D + x_G &\geq 1 \\
x_D + x_H &\geq 1 \\
x_E + x_F &\geq 1 \\
x_F + x_G &\geq 1 \\
0 \leq x_A, \ldots, x_H &\leq 1 \\
\min\ 24x_A + 6x_B + 10x_C + 12x_D &+ 30x_E + 50x_F + 15x_G + 18x_H.
\end{aligned}
$$

Ezt beadva kedvenc LP solverünknek azt kapjuk, hogy egy optimális megoldás az $x_B = x_D = x_F = 1$, többi változó 0, itt a célfüggvény értéke pedig 68; mivel itt egészértékű lett az optimum, így ez az eredeti, egészértékű feladatnak is egy optimális megoldása. Ha nem egészértékűt kaptunk volna, akkor is követhetjük pl. azt a heurisztikát, hogy olyan éleket próbálunk meg kiválasztani az LP feladat megoldása után, melynek egyik (vagy mindkét) végpontja az LP optimumában egy nemnulla súlyú csúcs.