Algoritmuselmélet vizsga
2012. december 20.

A rendelkezésre álló munkaidő 100 perc.
Az indoklás elengedhetetlen része a megoldásnak. Indoklás nélküli megoldásra nem jár pont.
Az eredményeket közzétesszük a www.cs.bme.hu/algel weboldalon, várhatóan ma este.
Minden feladat 10 pontot ér, a ponthatárok: 0-31: elégtelen, 32- 43: elégséges, 44- 55: közepes, 56-67: jó, 68-80: jeles.
Kiosztás és opcionális szóbeli holnap, december 21-én, 14.00-kor az IB 139-ben.

1. Milyen alakú inputok rendezésére alkalmas a radix rendezés? Írja le a radix rendezés algoritmusát és bizonyítsa be, hogy jól rendez! (A ládarendezést nem kell leírni.)

Egy lehetséges megoldás:

A radix rendezés olyan inputok rendezésére alkalmas, amelyek k komponensből állnak, ahol a komponensek lehetséges értékei egy-egy véges rendezett halmazból valók. A radix rendezés az elemeket a komponensek szerint lexikografikusan rendezi.

Az algoritmus először a k. koordináta szerint ládarendez, majd a $(k-1)$., $(k-2)$., $\ldots$, 1. szerint, ebben a sorrendben. Mindig az egész listát rendezzük, ügyelve rá, hogy azonos érték esetén a korábbi sorrendet megtartjuk (konzervatív rendezés).

Ekkor azt állítjuk, hogy a lista rendezve van.
Az állítás bizonyítása:
Két tetszőleges elem legyen $b = (b_1, b_2, \ldots, b_k)$ és $c = (c_1, c_2, \ldots, c_k)$, ahol $b < c$. Az állítás szerint a rendezés befejeztével b előrébb van, mint c.
$b_i = c_i \ \forall i$ esetén, amire $1 \leq i < j$, és legyen $b_j < c_j$, ahol $1 \leq j \leq k$
A radix rendezés előbb a k., majd a $(k-1)$., $\ldots$, $(j+1)$. koordináta szerint ládarendez.
A j. koordináta szerinti ládarendezés során a b a c elé kerül.
A továbbiakban a $(j-1)$., $\ldots$, 1. koordináta szerinti rendezés nem változtat a kettőjük sorrendjén, tehát a rendezés befejeztével b a c előtt lesz, így az állítást igazoltuk. □

---

2. Ebben a feladatban a piros-kék algoritmussal kapcsolatos kérdésekre kell válaszolnia. Mit jelent az, hogy egy részleges színezés takaros? Mondja ki a piros szabályt és magyarázza el, hogy a Kruskal algoritmusban hol és hogyan használjuk ezt a szabályt!

Egy lehetséges megoldás:

Egy színezésre akkor mondjuk hogy takaros, ha létezik olyan minimális feszítőfa, ami minden kék élet tartalmaz, de egyetlen piros élet sem tartalmaz.

Piros szabály: a gráfban válasszunk olyan kört, amiben nincs pirosra színezett él. Ekkor ebben a körben a legnagyobb súlyú élet színezzük pirosra.

A Kruskal algoritmusban az élek színezésekor akkor festünk egy élet pirosra, ha a soron következő befestendő él két vége ugyanabban a kék fában található.

---

3. Mi a Ládapakolás optimalizálási feladat? Írja le a megoldására tanult First Fit illetve First Fit Decreasing közelítő algoritmusokat! Fogalmazza meg a feladatot eldöntési problémaként is!

Egy lehetséges megoldás:

A Ládapakolás optimalizálási feladatnál a cél meghatározni, hogy adott n darab tárgyat, melyek egyesével $s_1, s_2, \ldots, s_n$ méretűek, hogyan osszuk szét a lehető legkevesebb ládába, ha egy láda maximális térfogat kapacitása 1. (Az elemeket és a ládát ,,egydimenziósnak" tekintjük, azaz csak egy irányban számít a méretük.)

A First Fit közelítő algoritmus a tárgyakon azok sorrendjében beleteszi az első olyan ládába, ahova még belefér a már benne levők mellé. Ha kell, akkor egy új, üres ládába rakja.

A First Fit Decreasing algoritmus esetén a tárgyakat először méretük szerint csökkenő sorrendbe rendezni, majd ugyanazt csinálja, amit A First Fit.

A feladat eldöntési problémaként megfogalmazva:

Input: $s_1, s_2, \ldots, s_n$ recionális számok; k egész szám.

Kérdés: Belefér-e k darab 1 kapacitású ládába az $s_1, s_2, \ldots, s_n$ méretű tárgy?

---

4. Dr. Watson egy n fokból álló lépcső tetejére szeretne feljutni, a lépcső fokai véletlenszerűen fehérre vagy feketére vannak festve. (Mindegyik fok be van festve valamelyik színnel, a lépcső teteje fehér.) Dr. Watson egyszerre legfeljebb $2k$ fokot tud lépni felfele, de háborús sérülése miatt könnyen kifárad, ezért bármely két szomszédos lépése közül legalább az egyik k-nál nem nagyobb. Adjon algoritmust, ami $O(nk)$ lépésben meghatározza, hogy legalább hány lépésre van szüksége, hogy feljusson lépcső tetejére úgy, hogy végig fehér színű fokokat használ!

Egy lehetséges megoldás:

A feladatot dinamikus programozás segítségével oldhatjuk meg.

Vegyük úgy, hogy kezdetben Dr. Watson a lépcső alján a 0. lépcsőfokon áll.

Ekkor jelölje $\mathcal{E}[i]$ az i. lépcsőfokra lépés minimális lépésszám-igényét (továbbiakban: költség) azon feltétel mellett, hogy (a 0. lépcsőfok kivételével, ahol kezdetben áll) csak fehérre festett lépcsőfokokra léphet és j lépcsőfokot lépett utoljára, ahol $1 \leq j \leq k$.

Hasonló feltételek mellett legyen $\mathcal{F}[i]$ az i. lépcsőfokra lépés minimális lépésszám-igénye, ha az i. lépcsőfokra történő rálépéskor j lépcsőfokot lépett át, és $k+1 \leq j \leq 2k$.

Továbbá legyen $\mathcal{E}[0] = 0$, és $\mathcal{F}[i] = \infty$, ha $i \leq k$.

Ekkor az $\mathcal{E}[i]$ kiszámítására $i > 0$ feltétel mellett az alábbi rekurziós összefüggés alkalmazható:

$$\mathcal{E}[i] = \begin{cases} \infty & \text{ha az } i. \text{ lépcsőfok fekete} \\ 1 & \text{ha az } i. \text{ lépcsőfok fehér és } i < j \\ \min\limits_{\forall j \in [1,k], i \geq j} \{\mathcal{E}[i-j]; \mathcal{F}[i-j]\} + 1 & \text{egyébként} \end{cases}$$

Az $\mathcal{F}[i]$ az alábbiak szerint adódnak ($i > k$ esetre):

$$\mathcal{F}[i] = \begin{cases} \infty & \text{ha az } i. \text{ lépcsőfok fekete} \\ 1 & \text{ha az } i. \text{ lépcsőfok fehér és } i \leq 2k \\ \min\limits_{\forall j \in (k,2k]} \{\mathcal{E}[i-j]\} + 1 & \text{ha } 2k < i \text{ és } 0 \leq i-j \end{cases}$$

A rekurziós összefüggések magyarázata: az i. lépcsőfokra lépéskor az i.-et megelőző $i-j$. lépcsőfokra való feljutás minimális költségét keressük (a ∞ szimbólum jelentése, hogy az adott

lépcsőfokra nem lehet feljutni), majd ezt növeljük 1-gyel, ami az utolsó lépés költségének figyelembe vételét jelenti. Az $\mathcal{E}$ és $\mathcal{F}$ vektorokra azért van szükség, hogy meg lehessen különböztetni azokat az eseteket, amikor egy adott lépcsőfokra k-nál nagyobb lépéssel jutottunk el, ilyenkor ezután csak legfeljebb k nagyságú lépés következhet.

Az algoritmus végrehajtása során ki kell számolni $\mathcal{E}[1], \mathcal{E}[2], \ldots, \mathcal{E}[n]$ és $\mathcal{F}[1], \mathcal{F}[2], \ldots, \mathcal{F}[n]$ értékeket.

Megoldás: $\min\{\mathcal{E}[n]; \mathcal{F}[n]\}$

Lépésszám: $O(nk)$, mivel minden lépcsőfokra (a számuk $O(n)$) a k és $2k$ lépéskorlátnak megfelelően meg kell határozni az előző k, illetve $2k$ lépcsőfok alapján ($O(2k) = O(k)$) a feljutás minimális költségét.

---

5. Egy piros-fekete fában n elemet tárolunk. Javasoljon olyan algoritmust, ami inputként megkapva a fát, a fában tárolt elemekből összehasonlítások használata nélkül felépít egy kupacot!

Egy lehetséges megoldás:

A kupac tulajdonságai, hogy a kupac egy teljes bináris fa, illetve minden S elemre igaz, hogy $S \leq$ *jobbfiú(S)* és $S \leq$ *balfiú(S)*. Ezt felhasználva világos, hogy növekvő sorrendbe rendezett elemekből a legegyszerűbb kupacot építeni, hiszen nincs más teendő, mint a legkisebb elemet felvenni a gyökérbe, és a többi elemet is a köztük felállított sorrendnek megfelelően beszúrni a kupacba. Ekkor a (speciálisan elvégzett) beszúrásnál az elem egyből jó helyre is kerül, hiszen nagyság szerint egyre nagyobb elemeket szúrunk be, és nem kell összehasonlítás, mert tudjuk, hogy nagyobb, mint a fában lévő bármely csúcs.

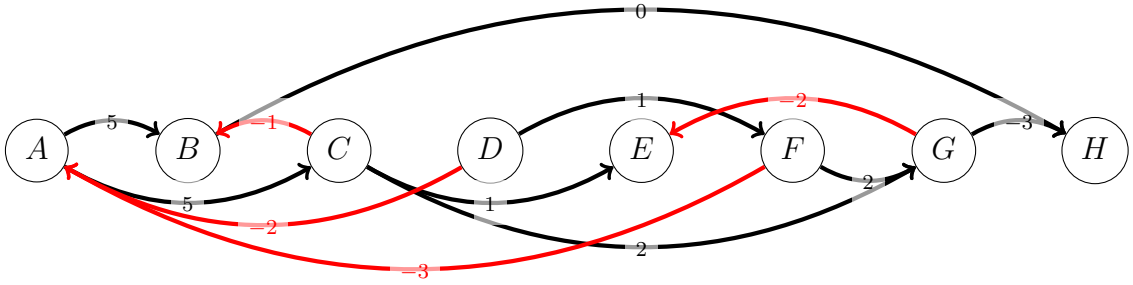
Feladat tehát előállítani a piros-fekete fában tárolt elemek nagyság szerint növekvő sorrendjét összehasonlítás alkalmazása nélkül. Erre egy megfelelő algoritmus a fa inorder bejárása. A leveleiben nem tárolunk elemket, így azokat figyelmen kívül hagyva a bejárást elvégezve az elemek nagyság szerint növekvő sorrendben állnak majd rendelkezésre. Ennek magyarázata, hogy a piros-fekete fa teljesíti a keresőfa tulajdonságot, azaz egy adott S elem esetén S bal részfájába eső elemek mindegyike kisebb, míg a jobbrészfába eső elemek mindegyike nagyobb, mint S. Az inorder bejárás összehasonlítások nélküli, az inorder fonál segítségével megkapható.

---

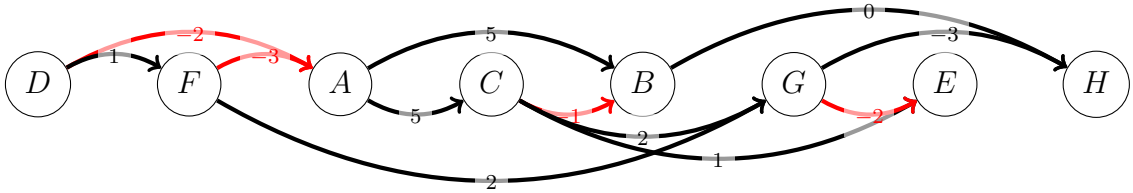
6. (a) Adja meg az alábbi, irányított kört nem tartalmazó gráf egy topologikus sorrendjét! A gráf éllistája (zárójelben az él súlya): **A**:$B(5), C(5)$; **B**:$H(0)$; **C**:$B(-1), G(2), E(1)$; **D**:$A(-2), F(1)$; **E**:$-$, **F**: $A(-3), G(2)$; **G**:$H(-3), E(-2)$, **H**: $-$.
(b) Határozza meg a topologikus sorrend segítségével a legnagyobb összhosszúságú, A-ból induló gráfbeli út hosszát és magát a leghosszabb utat is!

Egy lehetséges megoldás:

(a) A feladatban adott gráf lerajzolva:



Az ábrán pirossal berajzolt élek miatt a csúcsok ABC szerinti sorrendje nem egy topologikus rendezése a gráfnak. Az A, B, E csúcsok, és az A-ból elérhető C áthelyezésével egy helyes topologikus sorrend az alábbi (pirossal ugyanazok az élek vannak kiemelve, amik a fenti ábrán is pirossal szerepelnek):



Megjegyzés: A feladatot bonyolultabb esetben egy tanult tétel alkalmazásával lett volna érdemesebb megoldani, amely azt mondja ki, hogy DAG-ban, egy tetszőleges csúcsból indulva mélységi bejárást futtatva, majd a bejárásban kapott befejezési számok alapján fordított sorrendben leírva a csúcsokat egy topologikus rendezést kapunk.

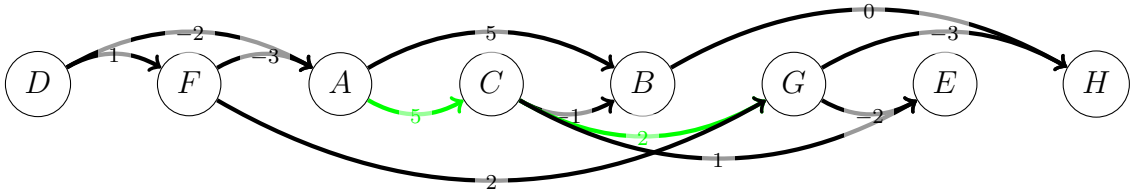
(b) A leghosszabb út keresésének csak akkor van értelme egy gráfban, ha az egy DAG (ez a tulajdonság most megvan). Az általános algoritmus erre az alábbi rekurzív képletet használja, ami s csúcsra (topologikus sorrend szerint haladva dinamikus programozással) kiszámolja a leghosszabb utat s-ből indulva:

$$\begin{aligned} d(s,s) &= 0 \\ d(s,v_i) &= \max_{(v_j,v_i)\in E}\{d(s,v_j)+c(v_j,v_i),0\} \end{aligned}$$

Ez alapján a kapott leghosszabb út meghatározása:

$D : 0$
$F : \max\{0+1;0\} = 1$
$A : \max\{-2;1+(-3);0\} = 0$
$C : \max\{0+5;0\} = 5$
$B : \max\{0+5;5+(-1);0\} = 5$
$G : \max\{1+2;5+2;0\} = 7$
$E : \max\{5+1;7+(-2);0\} = 6$
$H : \max\{5+0;7+(-3);0\} = 5$

A leghosszabb út tehát G-nél végződik és 7 hosszú, ez C-nél lévő 5-ös érték eredménye, amely pedig az A-nál lévő 0-é. A leghosszabb út lenti ábrán zölddel jelölt $A \to C \to G$ út, hossza 7.



7. Tegyük fel, hogy $NP \subseteq P$. Következik-e ebből, hogy az alábbi eldöntési feladat $coNP$-beli?

   Input: G irányítatlan gráf

   Kérdés: Igaz-e, hogy G élei kiszínezhetők 2014 színnel úgy, hogy az egy csúcsra illeszkedő élek színei mind különbözőek?

Egy lehetséges megoldás:

A megoldás során azt használjuk ki, hogy ha a problémáról belátjuk, hogy NP-beli, akkor $NP \subseteq P$ állításból következik, hogy P-beli is. Ekkor azonban, mivel tudjuk, hogy $P \subseteq coNP$, azaz a feladat kérdésére a válasz ebben az esetben az, hogy igen, következik.

Ehhez azonban be kell látni, hogy a feladatban adott eldöntési probléma (nevezzük L-nek) NP-beli. Erre egy hatékony tanúsítvány egy ilyen színezés. Hatékony, mert minden csúcsra megnézve, hogy az ott találkozó élek különböző színűek-e ($O(ne)$), illetve az éleket is megvizsgálva, hogy 2014 színt használtak-e a színezéshez ($O(e)$) polinom időben beláthatjuk a helyességét.

Megjegyzés: Jelen esetben triviális, de bonyolultabb esetben kellene annak bizonyítása is, hogy ténylegesen *tanúsítvány*. Azaz, hogy pontosan akkor létezik, amikor a válasz igen. Egy példa, hogy erre a feltételre miért is van szükség: az ,,igaz-e, hogy G-ben nincs Hamilton-kör" eldöntési probléma esetén nem tanúsítvány a csúcsok listája, amelyeket elvéve túl sok komponensre esik szét a gráf (lásd: BSZ), pedig *hatékony*. (Pedig ezzel bizonyítanánk, hogy $P = NP$, mert a fenti probléma H komplementere, tehát $coNP$-teljes.) Ennek oka, hogy a feltétel szükséges, de nem elégséges.

Tehát az L eldöntési probléma NP-beli, azaz a feltételezésnek megfelelően P-beli is, amiből pedig következik, hogy $coNP$-beli.

(Mivel nem NP-teljesség belátása a feladatunk, így hatékony tanúsítványon túl az NP-nehézség vagy a feladat polinomiális mivoltának bizonyítása nem szükséges.)

---

8. Igazolja, hogy a következő eldöntési probléma P-ben van, vagy azt, hogy NP-teljes!

   Input: G irányítatlan gráf

   Kérdés: Igaz-e, hogy van G-nek olyan feszítőfája, amelyben van egy pontosan 2012 fokú csúcs, minden más csúcs fokszáma pedig legfeljebb kettő?

Egy lehetséges megoldás:

A probléma NP-teljességét fogjuk megmutatni.

Jelöljük a feladatban szereplő eldöntési problémát L-lel. Elsőnek lássuk be, hogy L NP-beli egy hatékony tanúsítvány mutatásával. Jó hatékony tanúsítvány egy, a feltételeknek eleget tevő feszítőfa mutatása, hiszen az n csúcson egyesével végigjárva és megállapítva a fokszámukat ($O(n)$), végignézve, hogy léteznek-e az élek ($n-1$ kell legyen, $O(e)$), és megnézve, hogy

összefüggő-e/körmentes-e (DFS - $O(n+e)$) ellenőrizhető a helyessége $O(n)$ lépésben ($O(e) = O(n)$ fák esetén, ahol n a csúcsok, e pedig az élek száma).

Belátjuk, hogy H-út $\prec L$. A Karp-redukció során rendeljük a G gráfhoz azt a következő G' gráfot. Vegyünk fel egy új u csúcsot és a $w_1, w_2, \ldots, , w_{2011}$ csúcsokat. Kössük össze u-t minden $v \in V(G)$ csúccsal és minden w_j csúccsal. Ez a hozzárendelés polinom időben végrehajtható, mivel összesen 2012 új csúcsot és $|V(G)| + 2011$ új élet kell felvenni a gráfhoz.

Ekkor G'-ben pontosan akkor lesz olyan feszítőfa, aminek egyetlen 2012 fokszámú csúcsa van és a feszítőfában a többi csúcs fokszáma maximum 2, ha G tartalmaz Hamilton-utat.

Ha G-ben van Hamilton-út, akkor jelöljük egyik végpontját v_1-el. Ekkor a Hamilton-út éleihez hozzávéve a v_1u élet és az összes uw_j élet épp megfelelő feszítőfát kapunk.

Ha G'-ben van megfelő feszítőfa, akkor abban u 2012 fokú kell, hogy legyen, különben a feszítőfa nem tartalmazhatná az összes w_j pontot. Emiatt az uv_i élek közül pont egyet tartalmaz a feszítőfa. Ekkor a feszítőfa G-ben levő részében minden pont foka legfeljebb 2, ezért ez egy Hamilton-út G-ben.

G-ből G' származtatása, így a probléma visszavezetése polinom időben történt, tehát ezzel beláttuk, hogy a probléma NP-nehéz.

Mivel L NP-beli és NP-nehéz, így NP-teljes is.