

# Algel II. gyakorlat

## Dinamikusak vagyunk

2013. február 18.

1. Oldjuk meg a hátizsák-problémát (azaz töltsük ki a megfelelő táblázatot) az alábbi konkrét esetben:

Legyen  $n = 4$ , a tárgyak súlyai  $s_1 = 7, s_2 = 5, s_3 = 4, s_4 = 1$  és értékei  $v_1 = 20, v_2 = 14, v_3 = 10, v_4 = 1$ , a súlykorlát  $b = 10$ .

2. **[ZH: 2008. március 28.]** Egy  $n \times n$  méretű táblázat minden eleme egy egész szám. A táblázat bal alsó sarkából akarunk eljutni a jobb felső sarkába úgy, hogy egy lépésben a táblázatban vagy felfelé vagy jobbra egyet lépünk. Azt szeretnénk, hogy a lépegetés során látott elemek növekvő sorrendben kövessék egymást. Egy ilyen út értéke a benne szereplő számok összege. Adjon  $O(n^2)$  futási idejű algoritmust, ami meghatározza, hogy az adott táblázatban a szabályok szerinti utak értékei között mekkora a legnagyobb!
3. Adott egy fa, melynek csúcsaihoz súlyok vannak rendelve. Adjunk lineáris algoritmust, ami meghatározza a fában található maximális súlyú független ponthalmaz súlyát!
4. **[Vizsga: 2007. június 12.]** Egy  $n$  és egy  $m$  karakterből álló szövegben meg akarjuk találni a legnagyobb azonos darabot, azaz ha az egyik szöveg  $a_1 a_2 \cdots a_n$  és a másik  $b_1 b_2 \cdots b_m$ , akkor olyan  $1 \leq i \leq n$  és  $1 \leq j \leq m$  indexeket keresünk, hogy

$$a_{i+1} = b_{j+1}, a_{i+2} = b_{j+2}, \dots, a_{i+t} = b_{j+t}$$

teljesüljön a lehető legnagyobb  $t$  számra. Adjon erre a feladatra  $O(mn)$  lépést használó algoritmust!

- 
5. **[Vizsga: 2007. május 29.]** Legyen  $w = w_1 w_2 \cdots w_n$  egy  $n$  betűből álló szó. Hívjuk részsónak  $w$  egy tetszőleges  $w_i w_{i+1} \cdots w_{i+k}$  darabját ( $1 \leq i \leq n - 1, 1 \leq k \leq n - i$ ). Adjunk algoritmust, ami  $O(n)$  lépésben meghatározza az összes  $a$ -val kezdődő és  $b$ -re végződő részsó számát.
  6. Egy játékban egy  $n \times m$  rács bal felső sarkából kell eljutnunk a jobb alsó sarokba. Egy lépés során a rács mentén vízszintesen vagy függőlegesen tudunk a következő rácspontba lépni. Azonban van néhány kereszteződés, ahova nem szabad lépünk. Ezek helyét az  $R$  tömb írja le,  $R[i, j] = 1$ , ha az  $(i, j)$  kereszteződésbe nem léphetünk, egyébként  $R[i, j] = 0$ . Adjunk  $O(nm)$  futási idejű algoritmust annak meghatározására, hogy pontosan  $n + m - 2$  lépést téve a rácson hányféleképpen tudjuk a célt elérni!
  7. **[ZH: 2010. április 19.]** Egy  $n \times k$  méretű táblázatban van néhány megjelölt elem. A táblázat bal alsó sarkából akarunk eljutni a jobb felső sarkába úgy, hogy minden lépésben a táblázat egy eleméről vagy a közvetlen felette vagy a tőle jobbra lévő elemre mehetünk (ha van ilyen). Adjon  $O(nk)$  idejű algoritmust, amely a megjelölt elemek helyét ismerve meghatározza, hogy egy ilyen út során maximálisan hány alkalommal tudunk megjelölt elemre lépni!
  8. **[pZH: 2012. április 25.]** Egy  $n$ -szer  $n$ -es táblázat minden kockájába egy (nem feltétlenül pozitív) egész szám van írva vagy rózsaszínnel vagy sárgával. Sétát szeretnénk tenni a táblázatban a következő feltételekkel: a séta bárhol indulhat és bárhol végződhet, egy lépésben vagy balra vagy felfelé léphetünk egy mezőt, de csak akkor ha közben színt is váltunk. Egy ilyen séta értéke a közben érintett mezőkön lévő számok összege (a végponti mezők is számítanak). A séta állhat egy mezőből is, azaz végződhet ott, ahol kezdődik. Adjon algoritmust, ami  $O(n^2)$  lépésben meghatározza a táblázatban található legértékesebb séta hosszát.

9. Legyenek  $a_1, a_2, \dots, a_n$  tetszőleges egész számok és  $m < n^2$  egész. Adjunk algoritmust, amely a bináris alakjukkal megadott  $a_1, a_2, \dots, a_n$  és  $m$  számokról eldönti polinom időben, hogy az  $a_1, a_2, \dots, a_n$  számok közül kiválasztható-e néhány úgy, hogy az összegük  $m$ -mel osztva egyet adjon maradékul!
10. Adjunk algoritmust, ami egy  $n$  csúcsú fában lineáris időben meghatározza a fában levő leghosszabb út hosszát!
11. **[Vizsga: 2009. június 11.]** A véges hosszú 0-1 sorozatok egy részét valamilyen szempontból jónak tekintjük, a többit rossznak. Van egy  $A$  algoritmusunk, mely adott  $n$  hosszú 0-1 sorozatról  $O(\log(n!))$  lépésben megmondja, hogy a sorozat jó vagy rossz.
- Adjon olyan eljárást, mely  $A$ -t felhasználva, ha adott egy  $m$  hosszú 0-1 sorozat,  $y = y_1 y_2 \cdots y_m$ , akkor eldönti, hogy  $y$  előáll-e néhány jó sorozat egymás után fűzéséből. Az eljárás lépésszáma összesen legyen  $O(m^4)$ .
- Segítség:*  $\log(n!) \approx O(n \log n) = O(n^2)$ .
12. Egy  $n$  szóból álló szöveget kell sorokra tördelni. A szöveg  $i$ -edik szava  $\ell_i$  karakterből áll, egy sor  $s$  karakter hosszú. Ha egy sor a szöveg  $i$ -edik szavával kezdődik és a  $j$ -edik szóval végződik, akkor az elválasztó szóközöket is figyelembe véve  $t = s - (\ell_i + \ell_{i+1} + \cdots + \ell_j + j - i)$  üres hely marad a sor végén. Egy ilyen sor hibája legyen  $t^2$ . A tördelés hibája a nem üres sorok hibáinak összege. Adjunk  $O(n^2)$  lépéses algoritmust egy legkisebb hibájú tördelés meghatározására! (A szavak sorrendje rögzített.)
13. Az  $1, 2, \dots, n$  számoknak adott két permutációja,  $x_1, \dots, x_n$  és  $y_1, \dots, y_n$ . A két sorozat egy közös részsorozata egy  $1 \leq i_1 < \cdots < i_k \leq n$ , és egy  $1 \leq j_1 < \cdots < j_k \leq n$  indexsorozattal adható meg, ahol  $x_{i_m} = y_{j_m}$  teljesül minden  $1 \leq m \leq k$  esetén. Adjunk  $O(n^2)$  lépésszámú algoritmust, ami az  $x$  és  $y$  sorozatokban meghatároz egy leghosszabb közös részsorozatot!
14. **[MSc felvételi 2009. tavasz]** Adott egy  $n$  és egy  $m$  hosszú 0-1 sorozat,  $a_1, a_2, \dots, a_n$ , illetve  $b_1, b_2, \dots, b_m$ . Ezek alapján egy  $T$  tömböt töltöttünk ki a következő módon:
- Ha  $0 \leq i \leq n$ , akkor  $T[i, 0] = 0$ . Ha  $0 \leq j \leq m$ , akkor  $T[0, j] = 0$ .
- Ha  $1 \leq i \leq n$  és  $1 \leq j \leq m$ , akkor  $T[i, j] = \begin{cases} T[i-1, j-1] + 1 & \text{ha } a_i = b_j \\ \max(T[i, j-1], T[i-1, j]) & \text{ha } a_i \neq b_j \end{cases}$
- Írja le, hogy mi a jelentése a  $T[i, j]$  értéknek! A két sorozatnak milyen tulajdonságát adja meg a  $T[n, m]$  érték?
15. Adottak  $M_1, M_2, \dots, M_n$  munkák  $H_1, H_2, \dots, H_n$  határidőkkel és  $P_1, P_2, \dots, P_n$  profitokkal. Mindegyik munka elvégzése pontosan 1 napunkba kerül (egy napon csak egy munkát végezhetünk el). Adjunk  $O(n^2)$  lépésszámú algoritmust, amely meghatározza, hogy mely munkákat vállaljuk el a profit maximalizálása érdekében!