

### 3. gyakorlat

#### Dinamikus programozás

1. 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  $1 \leq i_1 < \dots < i_k \leq n$ , és egy  $1 \leq j_1 < \dots < j_k \leq n$  indexsorozattal adható meg, ahol  $x_{i_m} = y_{j_m}$  teljesül minden  $1 \leq m \leq k$  esetén. Adjon  $O(n^2)$  lépésszámú algoritmust, ami az  $x$  és  $y$  sorozatokban meghatároz egy leghosszabb közös részsorozatot.
- 

2. Legyen  $w = w_1 w_2 \dots w_n$  egy  $n$  betűből álló szó. Hívjuk részsónak  $w$  egy tetszőleges  $w_i w_{i+1} \dots w_{i+k}$  darabját ( $1 \leq i \leq n-1$ ,  $1 \leq k \leq n-i$ ). Adjon algoritmust, ami  $O(n)$  lépésben meghatározza az összes  $a$ -val kezdődő és  $b$ -re végződő részszo számát.
  3. 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!
  4. Az  $n$  elemű  $A$  tömb egész számokkal (lehetnek negatív számok is) van feltöltve. Adjon algoritmust, ami meghatároz egy olyan  $(i, j)$ ,  $1 \leq i \leq j \leq n$  indexpárt, amire  $A[i] + A[i+1] + \dots + A[j]$  maximális. (Azaz keressük a legnagyobb, folytonosan előálló összeget.) Az algoritmus futási ideje legyen  $O(n)$ .
  5. Legyenek  $a_1, a_2, \dots, a_n$  tetszőleges egész számok és  $m < n^2$  egész. Adjon 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.
- 

6. Egy  $n$  és egy  $m$  karakterből álló szövegben meg akarjuk találni a legnagyobb azonos darabot, azaz ha az egyik szöveg  $a_1 a_2 \dots a_n$  és a másik  $b_1 b_2 \dots b_m$ , 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.

7. Legyenek  $a_1, a_2, \dots, a_n$  tetszőleges egész számok és legyen  $b$  is egész szám. Adjon algoritmust, amely a bináris alakjukkal megadott  $a_1, a_2, \dots, a_n$  és  $b$  számokhoz  $O(nb)$  időben megadja, hogy a  $b$  szám hányféleképpen áll elő az  $a_1, a_2, \dots, a_n$  számok közül néhány összegeként.
8. 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  $t = s - (\ell_i + \ell_{i+1} + \dots + \ell_j + j - i)$  üres hely marad a sor végén. Egy ilyen sor hibája legyen  $t^2$ . A tördelés hibája a nem üres sorok hibáinak összege. Adjon  $O(n^2)$  lépéses algoritmust egy legkisebb hibájú tördelés meghatározására! (A szavak sorrendje rögzített.)