Dinamikus programozás

Algoritmuselmélet
2. gyakorlat

2013. február 22.

1. 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 lépnünk. Ezek helyét az R tömb írja le, $R[i,j] = 1$, ha az (i,j) kereszteződésbe nem léphetünk, egyébként $R[i,j] = 0$. Adjon $O(nm)$ futási idejű algoritmust annak meghatározására, hogy pontosan $n+m-2$ lépést téve a rácson hányféleképpen tudjuk a célt elérni.

2. Az $1, 2, \ldots, n$ számoknak adott két permutációja, $x_1, \ldots, x_n$ és $y_1, \ldots, y_n$. A két sorozat egy közös részsorozata egy $1 \le i_1 < \ldots < i_k \le n$, és egy $1 \le j_1 < \ldots < j_k \le n$ indexsorozattal adható meg, ahol $x_{i_m} = y_{j_m}$ teljesül minden $1 \le m \le k$ esetén. Adjon $O(n^2)$ lépésszámú algoritmust, ami az x és y sorozatokban meghatároz egy leghosszabb közös részsorozatot.

3. Az $n \times n$ méretű tábla minden mezőjére egy pozitív egész szám van írva, az i-edik sorának j-edik elemére $A[i,j]$, ahol $0 \le i,j \le n$. Feladat, hogy az első oszlopból eljussunk az utolsó oszlopba úgy, hogy egy lépésben mindig a következő oszlopba lépünk, és azon belül, ha az i-edik sorban voltunk, akkor a következő lépésben vagy az $i-1 \pmod n$ vagy az $i+1 \pmod n$ számú sorba kerülhetünk. Adjon $O(n^2)$ lépésszámú algoritmust, ami meghatározza, hogy az első oszlop melyik eleméből induljunk, ha azt akarjuk, hogy a bejárt mezőkön levő számok összege minimális legyen (az utolsó oszlop bármelyik mezője lehet az utolsó olyan mező, amire rálépünk).

4. 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 balra egyet lépünk. Azt szeretnénk, hogy a lépegetés során látott elemek növekvő sorrendben kövessék egymást. Egy ilyen út értéke a benne szereplő számok összege. Adjon $O(n^2)$ futásidejű algoritmust, ami meghatározza, hogy az adott táblázatban a szabályok szerinti utak értékei között mekkora a legnagyobb.

5. 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!

6. Legyen $w = w_1 w_2 \ldots w_n$ egy n betűből álló szó. Hívjuk részszónak w egy tetszőleges $w_i w_{i+1} \cdots w_{i+k}$ darabját ($1 \le i \le n-1$, $1 \le k \le n-i$). Adjon algoritmust, ami $O(n)$ lépésben meghatározza az összes a-val kezdődő és b-re végződő részszó számát.

7. 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 $1 \le i \le n$ és $1 \le j \le m$ 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.

8. 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.

9. 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.

10. Egy n szóból álló szöveget kell sorokra tördelni. A szöveg i-edik szava ℓ_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.)