Közelítés, dinamikus programozás

1. A LÁDAPAKOLÁS feladatban legyenek a súlyok: $s_1 = 0{,}4$; $s_2 = 0{,}7$; $s_3 = 0{,}1$; $s_4 = 0{,}6$. Hajtsa végre erre
   (a) az FF algoritmust;
   (b) az FFD algoritmust!
   (c) Hány ládát használ az optimális pakolás?

2. A LÁDAPAKOLÁS feladatban legyen két súly 0.34 és 4 súly 0.33. Hajtsa végre erre
   (a) az FFD algoritmust!
   (b) Hány ládát használ az optimális pakolás?

3. Tekintsük a LÁDAPAKOLÁS problémának azt a speciális esetét, amikor minden súly 1/2 vagy 1. Igazolja, hogy ez a változat P-ben van!

4. Egy csomagküldő szolgálatnál csupa egyforma dobozba pakolják az elküldendő árut. Céljuk az, hogy a ládákban maradó üres helyet kitöltő anyagból minél kevesebbre legyen szükség. Valaki azt javasolja, hogy alkalmazzák az FF algoritmust a pakolásra. Igaz-e, hogy ez erre a problémára is egy 2-közelítő algoritmust ad?

5. Az UTAZÓÜGYNÖK problémára tekintsük a következő mohó algoritmust: tetszőleges csúcsból indulunk, és minden alkalommal a legkisebb olyan súlyú élen megyünk tovább, ami új csúcsba visz. Ha már minden csúcsban jártunk, akkor egy élen visszatérünk a kezdőcsúcsba.

   Tetszőlegesen adott $c \geq 1$ konstansra adjon meg olyan gráfot, ami mutatja, hogy ez nem c-közelítő algoritmus!

6. Adott egy n és egy m hosszú 0-1 sorozat, $a_1, a_2, \ldots, a_n$, illetve $b_1, b_2, \ldots, b_m$. 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 $T[i,j] = \begin{cases} T[i-1, j-1] + 1 & \text{ha } a_i = b_j \\ \max\{T[i, j-1], T[i-1, j]\} & \text{ha } a_i \neq b_j \end{cases}$

   Mi a jelentése a $T[i, j]$ értéknek! A két sorozatnak milyen tulajdonságát adja meg a $T[n, m]$ érték?

7. Adott n pozitív egész szám, $a_1, a_2, \ldots, a_n$. Az $n+1$ sorból és $b+1$ oszlopból álló T táblázat sorait 0-tól n-ig, oszlopait 0-tól b-ig indexeljük. Legyen $T[0, 0] = 1$ és $T[0, c] = 0$ minden $1 \leq c \leq b$ értékre. Adjon eljárást, ami a T többi mezőjét összesen $O(nb)$ lépés alatt kitölti úgy, hogy $T[i, c]$ értéke azt mutassa, hányféleképpen lehet az $a_1, a_2, \ldots, a_i$ számok közül néhány összegeként a c számot előállítani ($1 \leq i \leq n, 1 \leq c \leq b$).

8. Tekintsük az RH problémának azt a változatát, amikor adottak az $a_1, a_2, \ldots, a_n$ és a b egész számok, melyekre teljesül, hogy $0 < a_i < n^2$ minden $1 \leq i \leq n$ esetén. Kérdés, hogy van-e olyan $I \subseteq \{1, \ldots, n\}$, melyre $\sum_{i \in I} a_i = b$. Mutassa meg, hogy ez a változat P-ben van!

9. Egy $n \times n$ méretű táblázat mezőin lépkedünk a bal alsó sarokból a jobb felső sarokba úgy, hogy egy lépésben a táblázatban vagy felfelé vagy jobbra egyet lépünk, de van néhány „tiltott” mező, ahova nem léphetünk. Adjon egy dinamikus programozást használó eljárást, ami meghatározza, hogy hányféleképpen érhetünk célba!

Közelítés, dinamikus programozás

1. A LÁDAPAKOLÁS feladatban legyenek a súlyok: $s_1 = 0{,}4$; $s_2 = 0{,}7$; $s_3 = 0{,}1$; $s_4 = 0{,}6$. Hajtsa végre erre
   (a) az FF algoritmust;
   (b) az FFD algoritmust!
   (c) Hány ládát használ az optimális pakolás?

2. A LÁDAPAKOLÁS feladatban legyen két súly 0.34 és 4 súly 0.33. Hajtsa végre erre
   (a) az FFD algoritmust!
   (b) Hány ládát használ az optimális pakolás?

3. Tekintsük a LÁDAPAKOLÁS problémának azt a speciális esetét, amikor minden súly 1/2 vagy 1. Igazolja, hogy ez a változat P-ben van!

4. Egy csomagküldő szolgálatnál csupa egyforma dobozba pakolják az elküldendő árut. Céljuk az, hogy a ládákban maradó üres helyet kitöltő anyagból minél kevesebbre legyen szükség. Valaki azt javasolja, hogy alkalmazzák az FF algoritmust a pakolásra. Igaz-e, hogy ez erre a problémára is egy 2-közelítő algoritmust ad?

5. Az UTAZÓÜGYNÖK problémára tekintsük a következő mohó algoritmust: tetszőleges csúcsból indulunk, és minden alkalommal a legkisebb olyan súlyú élen megyünk tovább, ami új csúcsba visz. Ha már minden csúcsban jártunk, akkor egy élen visszatérünk a kezdőcsúcsba.

   Tetszőlegesen adott $c \geq 1$ konstansra adjon meg olyan gráfot, ami mutatja, hogy ez nem c-közelítő algoritmus!

6. Adott egy n és egy m hosszú 0-1 sorozat, $a_1, a_2, \ldots, a_n$, illetve $b_1, b_2, \ldots, b_m$. 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 $T[i,j] = \begin{cases} T[i-1, j-1] + 1 & \text{ha } a_i = b_j \\ \max\{T[i, j-1], T[i-1, j]\} & \text{ha } a_i \neq b_j \end{cases}$

   Mi a jelentése a $T[i, j]$ értéknek! A két sorozatnak milyen tulajdonságát adja meg a $T[n, m]$ érték?

7. Adott n pozitív egész szám, $a_1, a_2, \ldots, a_n$. Az $n+1$ sorból és $b+1$ oszlopból álló T táblázat sorait 0-tól n-ig, oszlopait 0-tól b-ig indexeljük. Legyen $T[0, 0] = 1$ és $T[0, c] = 0$ minden $1 \leq c \leq b$ értékre. Adjon eljárást, ami a T többi mezőjét összesen $O(nb)$ lépés alatt kitölti úgy, hogy $T[i, c]$ értéke azt mutassa, hányféleképpen lehet az $a_1, a_2, \ldots, a_i$ számok közül néhány összegeként a c számot előállítani ($1 \leq i \leq n, 1 \leq c \leq b$).

8. Tekintsük az RH problémának azt a változatát, amikor adottak az $a_1, a_2, \ldots, a_n$ és a b egész számok, melyekre teljesül, hogy $0 < a_i < n^2$ minden $1 \leq i \leq n$ esetén. Kérdés, hogy van-e olyan $I \subseteq \{1, \ldots, n\}$, melyre $\sum_{i \in I} a_i = b$. Mutassa meg, hogy ez a változat P-ben van!

9. Egy $n \times n$ méretű táblázat mezőin lépkedünk a bal alsó sarokból a jobb felső sarokba úgy, hogy egy lépésben a táblázatban vagy felfelé vagy jobbra egyet lépünk, de van néhány „tiltott” mező, ahova nem léphetünk. Adjon egy dinamikus programozást használó eljárást, ami meghatározza, hogy hányféleképpen érhetünk célba!