Dinamikus programozás, rendezések eleje

1. 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 $1 \le c \le b$ 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 ($1 \le i \le n, 1 \le c \le b$).

2. 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 $1 \le i \le n$ 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!

3. Adjon $O(n^2)$ lépésszámú, dinamikus programozást használó algoritmust, ami megtalálja egy n hosszú $a_1, a_2, \ldots, a_n$ számsorozatban a leghosszabb növekvő részsorozatot. Például a $10, 3, 5, 2, 7, 1, 18, 4, 12, 17, 6$ sorozatban a leghosszabb növekvő részsorozat a $3, 5, 7, 12, 17$.

4. Az $A[1:n]$ tömbben levő elemekről tudjuk, hogy $A[1] \ne A[n]$. Adjon $O(\log n)$ összehasonlítást használó algoritmust, amely talál egy olyan i indexet, hogy $A[i] \ne A[i+1]$.

5. Rendezze a $3, 12, 1, 34, 4, 6, 0$ számsorozatot beszúrásos rendezéssel! Hány összehasonlításra volt szükség?

6. Tudjuk, hogy az $a_1, \ldots, a_n$ sorozat olyan, hogy egy darabig növekszik, utána csökken. Adjon $O(n)$ összehasonlítást használó algoritmust, ami növekvő sorrendbe rendezi az elemeit!

7. Az eredetileg növekvő $a_1, \ldots, a_n$ sorozatban egy elem értéke megváltozott, de nem tudjuk melyik. Hogyan lehet $O(n)$ lépésben újra növekvő sorrendbe rendezni az elemeket?

8. Az $A[1:n]$ tömbben számokat tárolunk. Határozza meg $O(n \log n)$ lépésben

   (a) azokat az értékeket, amelyek egynél többször fordulnak elő;

   (b) a leggyakoribb értékeket (vagyis azokat, amelyeknél többször semelyik másik szám sem fordul elő a tömbben)!

9. Legyen adott egy egészekből álló $A[1:n]$ tömb valamint egy b egész szám. Egy olyan $i, j \in \{1, \ldots, n\}$ indexpárt keresünk, melyre $A[i] + A[j] = b$. Hogyan lehet ezt $O(n \log n)$ időben megoldani?

10. Az n méretű (nem feltétlenül rendezett) A tömb elemei különböző pozitív számok. Adjon algoritmust, amely meghatároz egy $1 \le k \le n$ számot és kiválaszt k különböző elemet az A tömbből úgy, hogy a kiválasztott elemek összege nem több mint k^3. Ha nincs ilyen k, akkor az algoritmus jelezze ezt a tényt. Az algoritmus lépésszáma legyen $O(n \log n)$.

11. Adott a síkon n pont, melyek koordinátái (a_1, b_1), $(a_2, b_2), \ldots, (a_n, b_n)$. Olyan $P = (x, y)$ pontot keresünk a síkon, amire az $\sum_{i=1}^{n}(|a_i - x| + |b_i - y|)$ összeg minimális. Adjon algoritmust, amely $O(n \log n)$ lépésben meghatároz egy ilyen P pontot!

12. Adott a számegyenesen n intervallum, $[a_1, b_1], \ldots, [a_n, b_n]$. Azt akarjuk tudni, hogy együtt milyen hosszú részt fednek le a számegyenesből (azaz, hogy mennyi $\cup_{i=1}^{n}[a_i, b_i]$ összhossza). Adjon $O(n \log n)$ lépéses algoritmust ennek a hossznak a meghatározására!

13. Adjon minél kevesebb összehasonlítást használó algoritmust, ami n elem közül megtalálja a legkisebbet és a legnagyobbat is!

14. Adjon minél kevesebb összehasonlítást használó algoritmust, ami n elem közül megtalálja a két legkisebbet!