Varianta 1 – 3 probleme la alegere

Problemele trebuie rezolvate folosind metoda Divide et Impera. Complexitatea algoritmilor trebuie justificată (în scris)

1. Se consideră un vector a cu n elemente distincte (numerotate de la 0), ordonate crescător. Implementaţi un algoritm de complexitate **O(log n)** pentru a determina, dacă există, un indice i cu a[i] = i (în calculul complexității algoritmului nu se consideră și citirea vectorului) **[2]pb.2.17** (**1p**).

date.in	**date.out**
7 -7 -1 0 2 3 5 7	5

2. Se dau un număr natural n şi doi vectori de lungime n reprezentând parcurgerile în postordine şi inordine ale unui arbore binar având mulțimea vârfurilor {1,2,…, n}. Construiţi în memorie arborele binar corespunzător secvenţelor (alocat dinamic) şi afişaţi parcurgerea acestuia în preordine, inordine şi postordine. În cazul în care datele de intrare nu sunt corecte (nu reprezintă postordine și inordinea unui arbore binar) se va afișa un mesaj corespunzător (complexitate medie **O(n log(n))**). (**2p**)

date.in	**date.out**
4 4 1 2 3 1 4 3 2	3 1 4 2 1 4 3 2 4 1 2 3
date.in	date.out
4 4 2 1 3 1 4 3 2	nu

Suplimentar *– implementaţi algoritmul astfel încât complexitatea să fie O (n)*

3. Se dau doi vectori a şi b de lungime n, respectiv m, cu elementele ordonate crescător. Propuneţi un algoritm cât mai eficient pentru a determina mediana vectorului obținut prin interclasarea celor doi vectori **O(log(min{n,m}))** https://leetcode.com/problems/median-of-two-sorted-arrays/ (**3p)**

date.in	**date.out**
10 2 4 6 8 10 12 14 16 18 20 3 1 23 25	12

4. Dată o mulţime de puncte în plan (prin coordonatele lor), să de determine cea mai apropiată pereche de puncte (se vor afişa distanţa şi punctele) http://infoarena.ro/problema/cmap **O(n log n)** **[1]** (**4p**)

Bibliografie

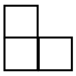
1. Jon Kleinberg, Éva Tardos, **Algorithm Design**, Addison-Wesley 2005 http://www.cs.princeton.edu/~wayne/kleinberg-tardos/
2. S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani, **Algorithms**, McGraw-Hill 2006
3. T.H. Cormen, C.E. Leiserson, R.R. Rivest – **Introducere în algoritmi**, MIT Press, trad. Computer Libris Agora

Varianta 2 – 3 probleme la alegere

Problemele trebuie rezolvate folosind metoda Divide et Impera, cu excepţia subpunctelor unde este precizat explicit altfel. Complexitatea algoritmilor trebuie justificată (în scris)

1. Se dă un arbore binar prin parcurgerea în preordine în care apar şi valorile null (semnificând lipsa unui fiu). Sa se construiască în memorie arborele (alocat dinamic) şi să se verifice dacă este arbore binar de căutare **O(n),** unde n=numărul de vârfuri (**1p**)

date.in	date.out
4 1 null 3 null null 7 6 null null null	da

2. Fie o tablă cu pătrăţele de dimensiune 2^nx2^n (n dat). Pe această tablă există o gaură la o poziţie dată prin linia şi coloana sa (lg, cg) (liniile şi coloanele se consideră numerotate de la 1). Pentru acoperirea acestei table avem la dispoziţie piese de forma 

   Aceste piese pot fi rotite cu 90°, 180° sau 270°. Să se afişeze o acoperire completă a tablei (cu excepţia găurii). Piesele vor fi reprezentate prin numere de la 1 la n, iar gaura prin 0 (cele 3 căsuţe ocupate de a i-a piesă pusă vor primi valoarea i) **O(2^{2n})** (**2p**)

date.in	date.out (un exemplu, soluţia nu este unică)
2 3 1	1 1 2 2 1 3 3 2 0 4 3 5 4 4 5 5

3. Se dau n valori distincte $x_1, x_2, \ldots, x_n$ şi ponderi asociate lor $w_1, w_2, \ldots$, respectiv $w_n \in [0, 1]$ cu $w_1+w_2 + \ldots + w_n = 1$. Să se determine mediana ponderată a acestor vectori, adică acea valoare x_k cu proprietăţile: $\sum_{x_i<x_k} w_i < 0,5$, $\sum_{x_i>x_k} w_i \le 0,5$. Justificaţi complexitatea algoritmului propus.

   De **exemplu**, pentru valorile

   x = 5 1 3 2 9 6 11 şi ponderile asociate

   w = 0,1 0,12 0,05 0,1 0,2 0,13 0,3

   mediana ponderată este x_k = 6, deoarece $\sum_{x_i<x_k} w_i = 0,12+0,1+0,05+0,1 = 0,37$, $\sum_{x_i>x_k} w_i = 0,5$

   Weighted median, problema 10-2 din [3] - complexitate caz mediu **O(n)** (**3p**)

date.in	date.out
7 5 1 3 2 9 6 11 0.10.12 0.05 0.1 0.2 0.13 0.3	6

   Suplimentar *– implementaţi un algoritm de complexitate O (n)*

4. Dată o mulţime de puncte în plan (prin coordonatele lor), să de determine cea mai apropiată pereche de puncte (se vor afişa distanţa şi punctele) http://infoarena.ro/problema/cmap **O(n log n) [1]** (**4p**)

Bibliografie

1. Jon Kleinberg, Éva Tardos, **Algorithm Design**, Addison-Wesley 2005 http://www.cs.princeton.edu/~wayne/kleinberg-tardos/

2. S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani, **Algorithms**, McGraw-Hill 2006

Varianta 3 – 3 probleme la alegere

Problemele trebuie rezolvate folosind metoda Divide et Impera. Complexitatea algoritmilor trebuie justificată (în scris)

1. Se dă un vector $a=(a_1,\ldots a_n)$ de tip munte (există un indice i astfel încât $a_1<a_2<\ldots<a_i > a_{i+1}>\ldots>a_n$; a_i se numește vârful muntelui). Propuneți un algoritm O(log n) care determină vârful muntelui (în calculul complexității algoritmului nu se consideră și citirea vectorului). **[1] exc 1,cap. 5** (**1p**)

date.in	**date.out**
5 4 8 10 11 5	11

2. http://www.infoarena.ro/problema/z (fără a memora tabla) **O(kn)** (**2p**)

3. Se consideră un vector cu n elemente. Se numeşte inversiune semnificativă a vectorului o pereche perechi (i, j) cu proprietatea că i < j şi $a_i > 2*a_j$. Să de determine **numărul** de inversiuni semnificative din vector. De exemplu, vectorul 4, 8, 11, 3, 5 are 3 inversiuni semnificative: (8,3), (11,3), (11,5) - **O(n log n)** **[1] exc. 2, cap. 5** (**3p**)

date.in	**date.out**
5 4 8 11 3 5	3

4. Dată o mulţime de puncte în plan (prin coordonatele lor), să de determine cea mai apropiată pereche de puncte (se vor afişa distanţa şi punctele). http://infoarena.ro/problema/cmap **O(n log n)** **[1]** (**4p**)

Bibliografie

1. Jon Kleinberg, Éva Tardos, Algorithm Design, Addison-Wesley 2005 http://www.cs.princeton.edu/~wayne/kleinberg-tardos/
2. S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani, Algorithms, McGraw-Hill 2006