Metoda Divide et Impera

Problemele trebuie rezolvate folosind metoda Divide et Impera. Complexitatea algoritmilor trebuie justificată

1. Se dă un vector a=(a_1,…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 numeşte vârful muntelui). Propuneți un algoritm O(log n) care determină vârful muntelui (în calculul complexității algoritmului nu se consideră şi citirea vectorului). **[1] exc 1,cap. 5**

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

2. **a)** Se citeşte de la tastatură un număr natural N. Se consideră o tablă (matrice) pătratică de dimensiuni 2^N*2^N pe care se scriu numerele naturale de la *1* si 2^N*2^N prin vizitarea **recursivă** a celor patru cadrane ale tablei în ordinea indicată şi în figura alăturată: dreapta-sus, stânga-jos, stânga-sus, dreapta-jos. De exemplu, daca *N=2*, tabla este completată astfel:

3	**1**
2	**4**

11 9 3 1
10 12 2 4
7 5 15 13
6 8 14 16

Să se afişeze în fişierul tabla.out matricea completată după regulile precizate.

intrare	**tabla.out**
2	11 9 3 1 10 12 2 4 7 5 15 13 6 8 14 16

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

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

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

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

4. 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}. Construiți în memorie arborele binar corespunzător secvențelor (alocat dinamic) şi afişați parcurgerea acestuia în preordine, inordine şi postordine. În cazul în care datele de intrare nu sunt corecte (nu reprezintă postordine şi inordinea unui arbore binar) se va afişa un mesaj corespunzător (complexitate medie **O(n log(n))**).

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

5. 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 + v. curs**

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

6. 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 valori, adică acea valoare x_k cu proprietățile: $\sum_{x_i<x_k} w_i < 0,5$, $\sum_{x_i>x_k} w_i \le 0,5$. Justificați 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) + v. curs**

date.in	date.out
7 5 1 3 2 9 6 11 0.1 0.12 0.05 0.1 0.2 0.13 0.3	6

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

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

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

8. Dată o mulțime de puncte în plan (prin coordonatele lor), să de determine cea mai apropiată pereche de puncte (se vor afişa distanța şi punctele) http://infoarena.ro/problema/cmap. Propuneți o implementare în care împărțirea planului (etapa de divide) să se facă printr-o dreaptă orizontală **O(n log n) [1] + v. curs**

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