Întrebări generale:

1. T(n) = T(n - 1) + T(n - 2) + n

Vezi aici.

Adunăm $n + 3$ **la relaţia de mai sus:** $T(n) + n + 3 = T(n-1) + n + 2 + T(n-2) + n + 1$

Notăm $S(n) = T(n) + n + 3$ **şi observăm, conform mai sus, că** $S(n) = S(n-1) + S(n-2)$

Ecuaţia caracteristică a formulei de recurenţă găsită este $x^2 = x + 1$ **cu o soluţie posibilă** $\lambda = \frac{1+\sqrt{5}}{2}$

Rezultă că $S(n) \in \Theta(\lambda^n)$**, deci şi** $T(n) \in \Theta(\lambda^n)$

Examen - Algoritmi si Structuri de Date Seria 14

3 Februarie 2017

1 Exercitii de nota 5 - (5 puncte)

1.1 0,5 puncte (0,25 puncte pe exercitiu)

Exprimati functiile urmatoare in notatia Θ:

(a) $n^3/2000 + n^2 \cdot 2^{100000} + 10000 \cdot n + 10$

Termen dominant: n^3 **, deci** $\Theta(n^3)$

(b) $\ln^2 n + \sqrt{n}$

Dacă nu ne dăm seama la prima vedere care este termenul dominant, putem aplica limită:

$$\lim_{n\to\infty} \frac{\ln^2 n}{\sqrt{n}} = (l'Hopital) \lim_{n\to\infty}\left(\frac{2\ln n}{n} \cdot 2\sqrt{n}\right) = 4 \cdot \lim_{n\to\infty} \frac{\ln n}{\sqrt{n}} =$$

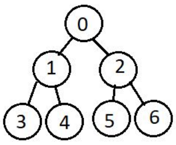
$$= (l'Hopital)\ 4 \cdot \lim_{n\to\infty} \frac{\frac{1}{n}}{\frac{1}{2\sqrt{n}}} = 8 \cdot \lim_{n\to\infty} \frac{\sqrt{n}}{n} = 8 \cdot \lim_{n\to\infty} \frac{1}{\sqrt{n}} = 0$$

Deci concluzionăm că $\sqrt{n}$ **creşte mai repede decât** $\ln^2 n$**, deci e termen dominant**

$\Theta(\sqrt{n})$

1.2 0,5 puncte

Desenati un arbore binar complet cu 7 noduri si scrieti matricea de adiacenta corespunzatoare.



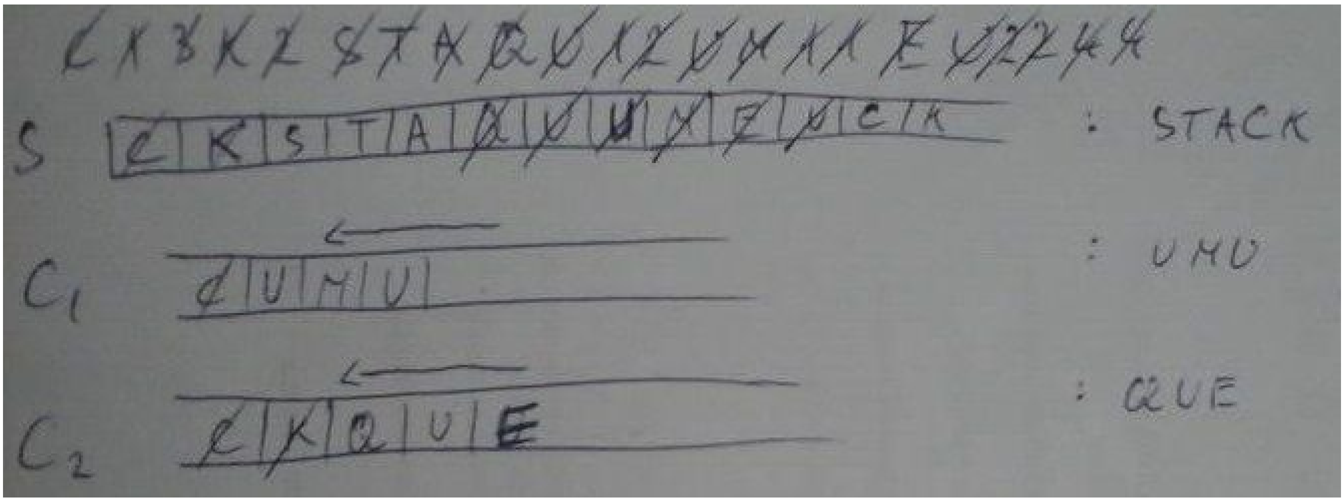
	0	1	2	3	4	5	6
0	-	1	1	0	0	0	0
1	1	-	0	1	1	0	0
2	1	0	-	0	0	1	1
3	0	1	0	-	0	0	0
4	0	1	0	0	-	0	0
5	0	0	1	0	0	-	0
6	0	0	1	0	0	0	-

1.3 1 punct (0,5 puncte pe exercitiu)

Se dau urmatoarele structuri de date: o stiva S si doua cozi C_1 si C_2 ce contin caractere. Cele trei structuri sunt initial vide si se considera de capacitate infinita. Se considera urmatoarele operatii:

X: se introduce caracterul "X" in S;
1: daca S e nevida, se extrage un element si se introduce in C_1, altfel nu se face nimic;
2 : daca S e nevida, se extrage un element si se introduce in C_2, altfel nu se face nimic;
3 : daca C_1 e nevida, se extrage un element si se introduce in C_2, altfel nu se face nimic;
4 : daca C_2 e nevida, se extrage un element si se introduce in S, altfel nu se face nimic. Cozile se considera cu capatul pentru inserare in dreapta si cel pentru stergere in stanga, iar stiva are capatul pentru inserare si stergere in dreapta.

(a) Sa se scrie continutul stivei S si al cozilor C_1 si C_2, dupa executarea urmatoarei secvente de operatii: C 1 3 K 2 S T A Q U 1 2 U N 1 1 E U 2 2 4 4



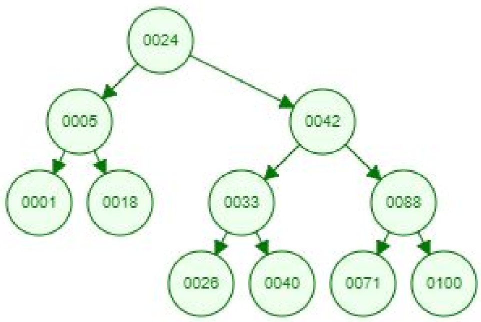
(b) Sa se scrie o secventa de operatii care are ca rezultat cuvantul “ROSU” in stiva S, cuvantul “VERDE” in coada C_2, iar C_1 este vida.

ROSUEDREV22222 sau ROSUV2E2R2D2E

1.4 1 punct (0,5 puncte pe exercitiu)

Se dau urmatoarele chei citite pe rand de la consola: 24, 5, 18, 42, 88, 71, 33, 1, 40, 26, 100

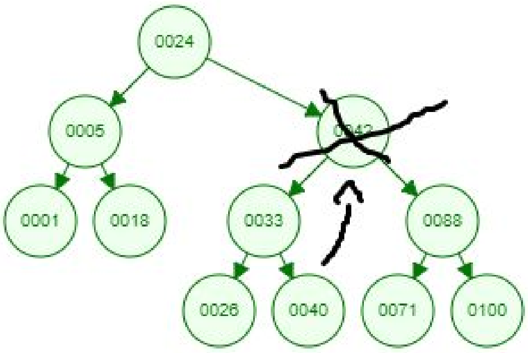
(a) Sa se construiasca arborele binar de cautare rezultat din inserarea lor pe rand, in ordinea citirii.

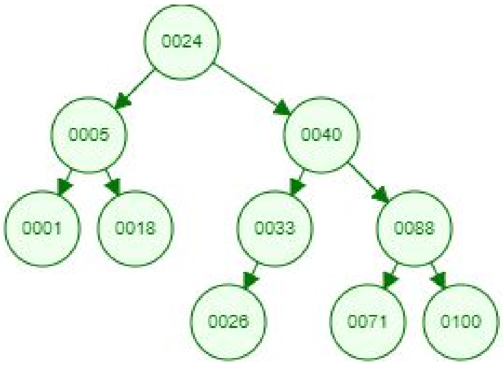


(https://www.cs.usfca.edu/~galles/visualization/BST.html)

(b) Sa se extraga din arborele construit cheia cu valoarea 42, ilustrandu-se arborele rezultat.

În momentul în care ştergem un nod, trebuie să-l înlocuim cu predecesorul (sau succesorul, depinde de convenţie) său în inordine ca să nu-şi strice proprietatea de arbore. În cazul de faţă, predecesorul lui 42 în inordine este 40.

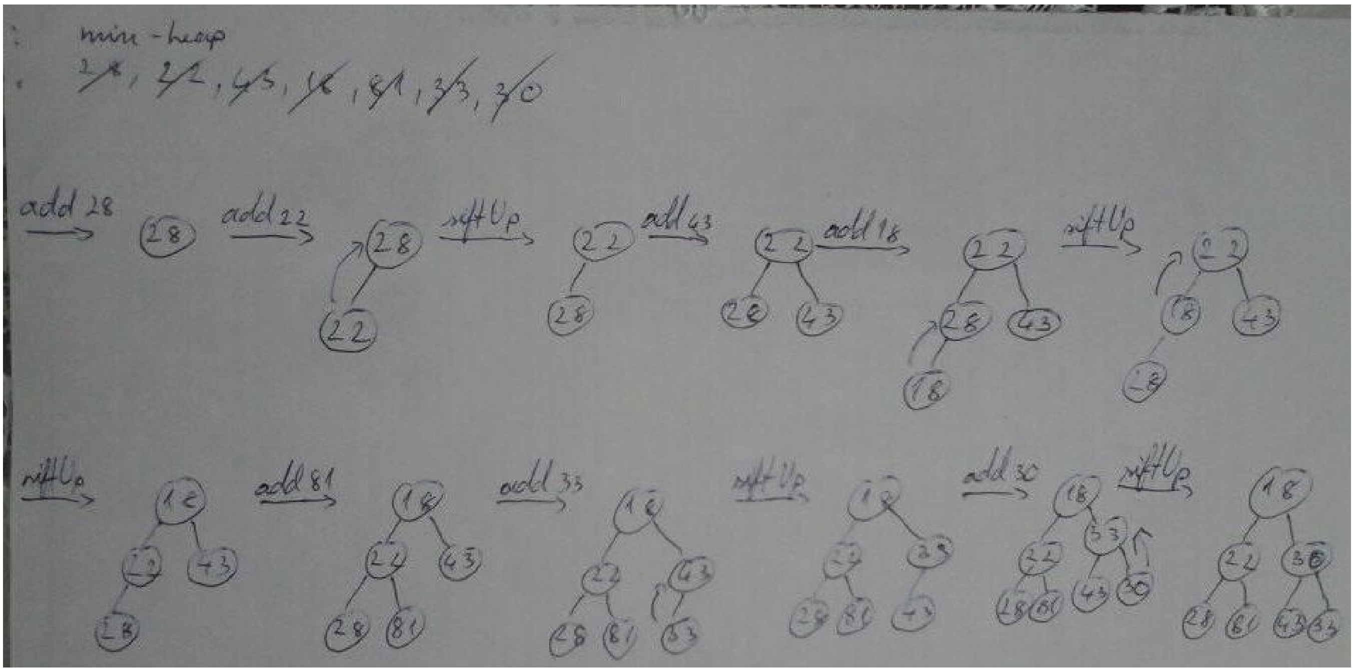




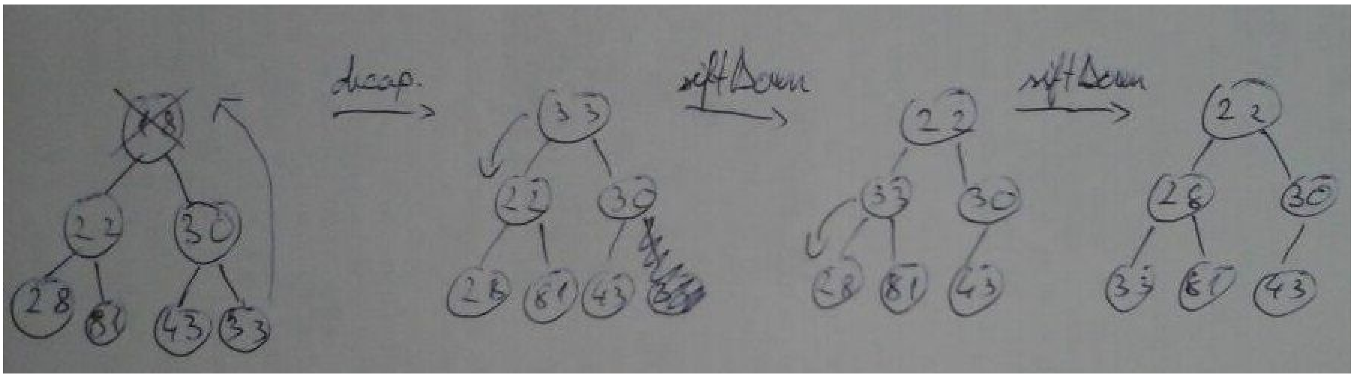
1.5 1 punct (0,5 puncte pe exercitiu)

Sa se construiasca heap-uri(ansamble) prin insertia pe rand a urmatoarelor chei (sa se ilustreze pasii intermediari, cu explicatii). Apoi, sa se extraga radacina din heap-uri-le rezultate.

(a) min-heap: 28, 22, 43, 18, 81, 33, 30

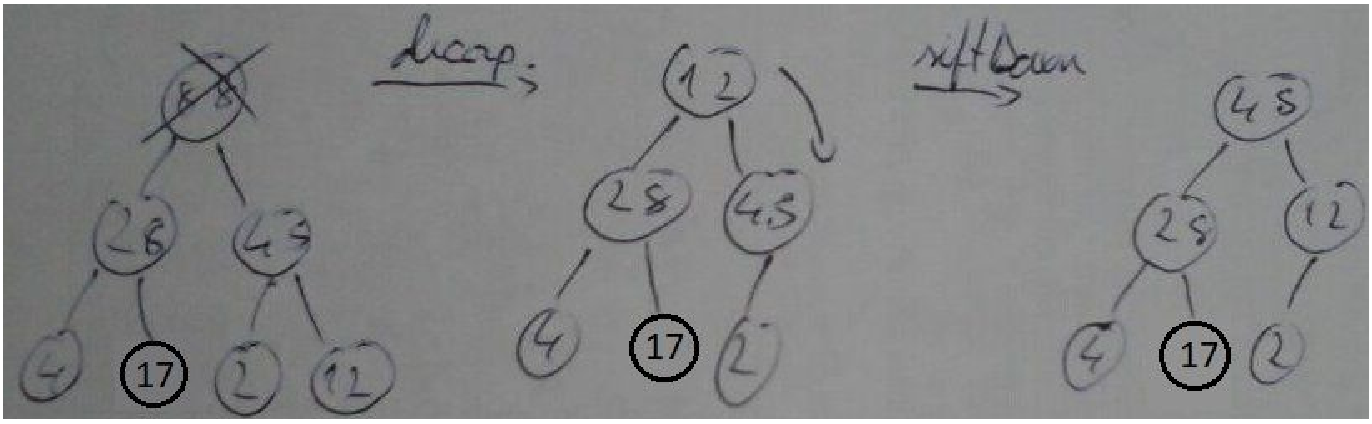


Extragerea minimului:



(b) max-heap: 12, 4, 28, 17, 45, 2, 88

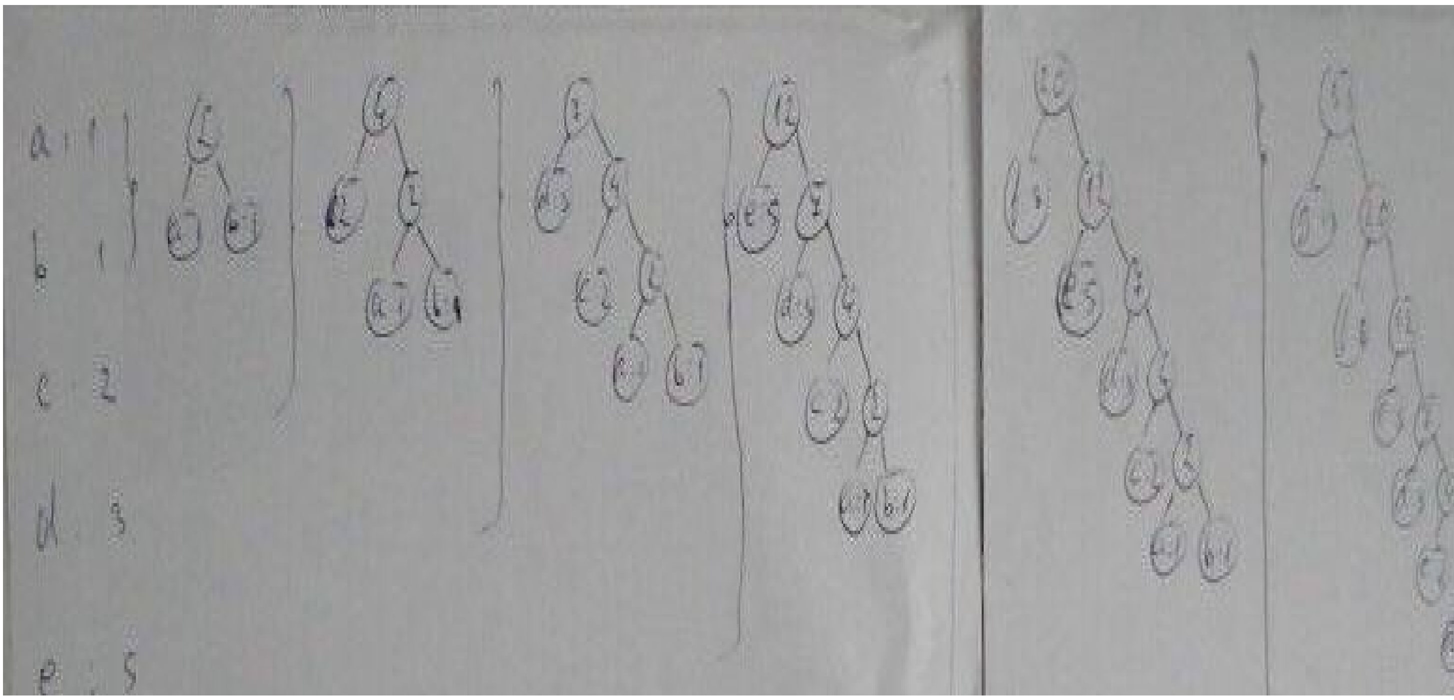
Extragerea maximului:



1.6 1 punct (0,5 puncte pe exercitiu)

(a) Care este codul (arborele) Huffman optim pentru urmatoarele frecvente, corespunzatoare primelor 8 numere Fibonacci: *a:1 b:1 c:2 d:3 e:5 f:8 g:13 h:21* ?

Daca exista mai multi arbori optimi, oricare din ei va primi punctajul maxim



(b) Generalizati raspunsul. Gasiti codul optim pentru un set de frecvente corespunzatorprimelor *n* numere Fibonacci. Este suficienta o descriere informala, fara demonstratie.

Observăm că arborii Huffman creaţi din numerele Fibonacci sunt într-o stare maximă de dezechilibru. Formula generală pentru codul celui de-al i-lea număr Fibonacci dintr-un arbore cu primele n numere Fibonacci este: 1..10, unde număul de "1"-uri este egal cu n-i

2 Exercitii cu demonstratii - (3 puncte)

2.1 1 punct

Rezolvati recurenta $T(n) = T(n-1) + n$ si demonstrati ca solutia gasita este corecta.

$T(n) = T(n-1) + n$ [1]

$T(n) = T(n-1) + n = T(n-2) + (n-1) + n = 0 + 1 + \ldots + (n-1) + n = \frac{n(n+1)}{2}$

$T(n) = \frac{n(n+1)}{2}$ [2]

Demonstrăm prin inducţie:

$T(k) \rightarrow T(k+1)$

$T(k+1) = T(k) + (k+1) = \frac{k(k+1)}{2} + (k+1) = \frac{k(k+1)}{2} + \frac{2(k+1)}{2} = \frac{(k+1)(k+2)}{2}$

$T(k+1) = (din\ [2])\ \frac{(k+1)(k+2)}{2}$

Am mers cu presupunerea că $T(k) = \frac{k(k+1)}{2}$ **şi am demonstrat că** $T(k+1) = \frac{(k+1)(k+2)}{2}$

2.2 1 punct

Demonstrati ca $\ln(n!) = \Theta(n \ln n)$

Vezi 2.2 de aici

2.3 1 punctz

Fie T un *arbore binar de cautare* si x un nod din arbore care *are doi copii*. Demonstrati ca succesorul nodului x nu are fiu stang, iar predecesorul lui x nu are fiu drept.

Vezi 2.3 de aici

3 Exercitii cu algoritmi - (3,5 puncte)

3.1 (1 punct)

Scrieti un algoritm (in pseudocod) care sa rezolve urmatoarea problema. Se da o multime S ce contine n numere naturale distincte si un numar natural x. Decideti daca numarul x poate fi exprimat ca suma de doua numere distincte din S.

Exemplul 1: S = {3,2,5,7,6} si x = 8. Raspunsul este DA. Exemplul 2: S = {3,2,5,6,7} si x = 4. Raspunsul este NU.

Pentru un algoritm cu timp de rulare $O(n^2)$ primiti 0,25 puncte. Pentru un algoritm de complexitate $O(n\log n)$ sau $O(n)$, veti primi punctajul intreg.

Cod în O(n):

```
#include <iostream>
#include <vector>
#include <map>

using namespace std;
//INPUT
vector<int> S = {3, 2, 5, 7, 6};
int x = 8;
////
int main() {
   map<int, bool> aparitii;
   //completăm map-ul de apariţii
   for (int i = 0; i < S.size(); i++)
       aparitii[S[i]] = true;
   //parcurgem S-ul
   for (int i = 0; i < S.size(); i++) {
       int a = S[i];
       int b = x - a;
       //vedem dacă există în vectorul de apariţii un nr b astfel încât a + b = x;
       if (aparitii[b] == true && b != a) {
           cout << "DA";
```

```
            return 0;
        }
    }
    cout << "NU";
}
```

Am ales un map de apariţii în loc de vector de apariţii ca să evit să scriu cod în plus pt iniţializarea de vector… dar ideea e aceeaşi.

Solutie in O(nlog n) : Se sorteaza numerele cu quick-sort.

Se parcurg numerele nr_i din S, iar pentru fiecare cautam binar (x-nr_i). i = 0 … n-1. In cazul i'n care cautarea binara se incheie cu succes, x se scrie ca (nr_i + x - nr_i). N.C

3.2 (1 punct)

Explicati cum se poate modifica metoda de sortare quicksort pentru ca aceasta sa ruleze in cazul cel mai defavorabil (i.e., worst-case) in timp $O(n\log n)$, presupunand ca toate numerele ce trebuie sortate sunt *distincte*.

Alegem pivotul ca fiind mediana. Algoritmul de determinare al medianei are complexitatea O(n).

Aceasta alegere ne garanteaza o "impartirea egala", astfel recurenta devine:

T(n) = 2* T(n/2) + O(n) , unde O(n) este timpul de determinare al medianei si T(n/2) timpul necesar rezolvarii fiecarei noi subprobleme. N.C

3.3 1,5 puncte

Fie $X[1 :: n]$ si $Y[1 :: n]$ doi vectori, fiecare continand n numere *sortate*. Prezentati un algoritm care sa gaseasca mediana celor 2n elemente. Mediana unei multimi de n elemente este elementul de pe pozitia $[n/2]$ in sirul sortat. De exemplu, mediana multimii 3,1,7,6,4,9 este 4.

In functie de timpul de rulare al algoritmului veti primi urmatoarele punctaje: $O(n\log n)$ - (0,25 puncte); $O(n)$ - (0,5 puncte); $O(\log^2 n)$ - (1 punct); $O(\log n)$ - (1,5 puncte).

Vezi aici (de la Prof. Marinescu Ruxandra)