

# BIBLIOGRAFIE TEME PENTRU LABORATOR

## (pentru EVALUARE)

ATENȚIE!!! FOARTE IMPORTANT!

Vă puteți folosi, pentru a învăța și înțelege rezolvarea, de rezolvările / implementările existente, dar VĂ CER să aveți (să vă formați) propriul stil (propria viziune, structură) de a PROGRAMĂ / REZOLVA O PROBLEMĂ.

Se poate obține cel mult nota 8 dacă rezolvarea o consider ca fiind preluată dacă (doar dacă) studentul știe / a înțeles în totalitate rezolvarea. Altfel rezolvarea nu va fi considerată. Nu accept contestații legate de decizia mea. Cu alte cuvinte, vreau să spun NU ÎNCERCAȚI SĂ COPIAȚI. SPOR!!!

### Bibliografie:

1. Thomas ~~Cormen~~ CORMEN and others, Introduction to ALGORITHMS, Editia a patra, Ed. The MIT Press, Cambridge, Massachusetts, London, England, 2022
2. Site-ul: [www.geeksforgeeks.org](http://www.geeksforgeeks.org)

## TEME PENTRU LABORATOR

( pentru evaluare)

LAB

1. Să se implementeze metodele de sortare:

a. Metoda selectiei -  $O(n^2)$

b. Metoda Heapsort -  $O(n \log_2 n)$

LAB

2. Să se implementeze operațiile indicate mai jos pentru liste simplu înălțuite, pentru liste dublu înălțuite, pentru liste circulare:

a. crearea unei liste

b. afișarea ~~ans~~ cheilor reținute într-o listă

c. inserarea unui nod într-o listă

d. căutarea unei chei într-o listă

e. ștergerea unui nod dintr-o listă

LAB

3.

3.1 Să se verifice corectitudinea introducerii de paranteze folosind o structură de date de tip stivă alocată dinamic. Presupunem că avem trei tipuri de paranteze:  $()$ ,  $[]$  și  $\{\}$ .

Indicație și observație:

a. Se admite ca fiind corectă expresia  $(\{\} [ ( ) ] )$ , deși matematic ordinea parantezelor este  $\{, [, \}$  și respectiv  $($ .

Nu este corectă expresia  $(\{ ( \} ) )$ , mai întâi trebuie să se închidă paranteza rotundă interioară și după aceea paranteza  $\}$ .

b. Structura de tip stivă alocată dinamic se va folosi precum urmează: cât timp extragem din expresia (care poate fi dată interactiv de la tastatură sau poate fi reținută într-un vector) care conține un sir de paranteze o paranteză deschisă, o

introducem într-o structură de tip stivă alocată dinamic; când extragem o paranteză închisă, verificăm dacă este paranteza corespunzătoare parantezei reținute în capul stive (vârful stivei) (adică paranteza „)”) pentru paranteza „(” ș.a.m.d.); dacă este eliminăm nodul / vârful stivei ș.a.m.d..

- 3.2 Pentru fiecare element dintr-un vector să se determine primul element mai mare decât el care urmează după el în vector folosind o stivă alocată dinamic. Să se introducă perechea formată din poziția la care se găsește elementul în vector și poziția la care se găsește elementul mai mare decât el (primul) într-o structură de tip coadă alocată dinamic.

Detalii / indicație:

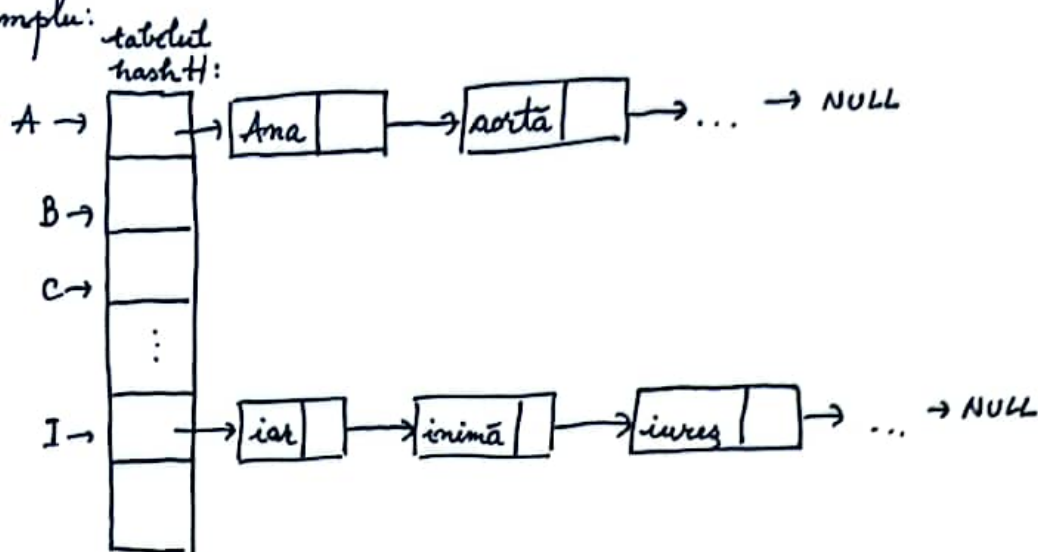
Pentru fiecare element din vector dacă pe următoarea poziție din vector se află un element cu o valoare mai mare decât elementul curent, perechea formată din pozițiile la care se găsesc aceste elemente se introduce în coadă, altfel poziția elementului curent se introduce în stivă. Dacă pentru un element găsim pe următoarea poziție o valoare mai mare și stiva nu este goală, se verifică dacă elementul aflat pe poziția următoare elementului curent este mai mare și decât elementele de stivă (până când se găsește în stivă un element mai mare decât el) se elimină aceste elemente din stivă și se introduce în coadă perechile cerute.



LAB

4. Să se construiască (să se stocheze) un dicționar folosind structura de date „hash table”.

Exemplu:



$H[1]$  = reține un pointer spre o listă ce conține cuvinte cu litera „A/a” ordonate crescător lexicografic

LAB

5. Pentru rezolvarea aplicațiilor următoare se poate consulta site-ul:

[www.geeksforgeeks.org/binary-search-tree-data-structure](http://www.geeksforgeeks.org/binary-search-tree-data-structure)

- 5.1 Construiți un arbore binar de căutare echilibrat pornind de la un vector sortat dat.
- 5.2 Interclasarea a doi arbori binari de căutare echilibrați.
- 5.3 Studiați și implementați două dintre aplicațiile de la „Hard Standard Problems in BST” (bineînțeles, atâtea câte se cerute la punctele 5.1 și 5.2).

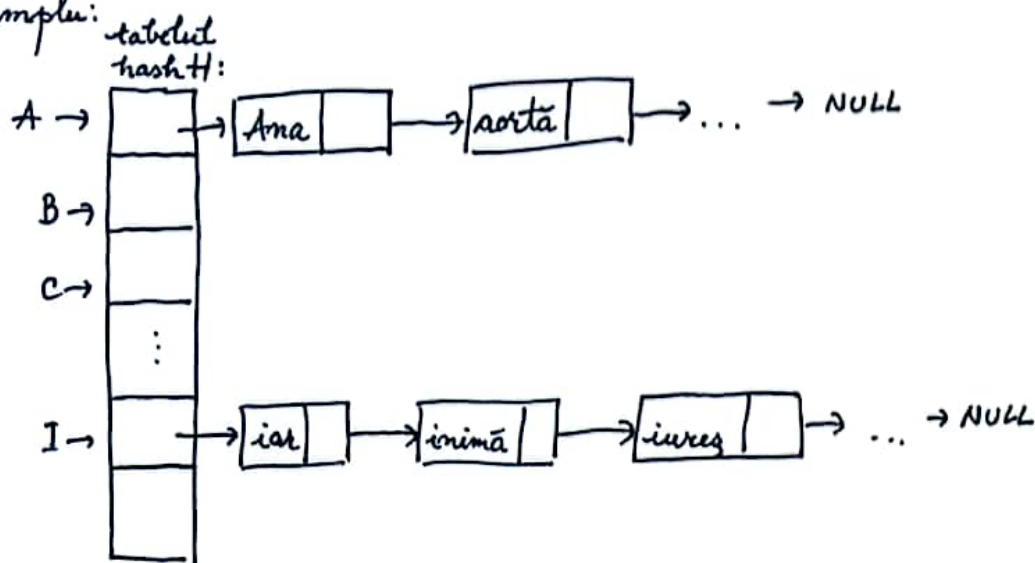
LAB

6. Implementați Algoritmii Kruskal și Prim.

## LAB

4. Să se construiască (să se stocheze) un dicționar folosind structura de date „hash table”.

Exemplu:



$H[1]$  = reține un pointer spre o listă ce conține cuvinte cu litera „A/a” ordonate crescător lexicografic

## LAB

5. Pentru rezolvarea aplicațiilor următoare se poate consulta site-ul:

[www.geeksforgeeks.org/binary-search-tree-data-structure](http://www.geeksforgeeks.org/binary-search-tree-data-structure)

- 5.1 Construiți un arbore binar de căutare echilibrat pornind de la un vector sortat dat.
- 5.2 Interclasarea a doi arbori binari de căutare echilibrați.
- 5.3 Studiați și implementați două dintre aplicațiile de la „Hard Standard Problems in BST” (bineînțeles, atâtea cât cele cerute la punctele 5.1 și 5.2).

## LAB

6. Implementați Algoritmii Kruskal și Prim.