

# Recapitulare pentru examen - iunie 2021

Universitatea "Transilvania" din Braşov

May 31, 2022

# Stive, cozi, liste

## Conținuturi teoretice

- Stive, cozi: - mod de funcționare, probleme care necesita stive/cozi
- Liste înlănțuite: - operații pe liste simple și dublu înlănțuite. Complexități.

## Stive, cozi, liste - Exemple de probleme

- 1 Se dă o listă înlănțuită  $L$ , care stochează numere întregi. Scrieți o funcție eficientă în pseudo-cod, care să elimine din listă toate dublurile. Ce complexitate are funcția dvs?

# Arbori binari

## Conținuturi teoretice

- noțiuni: noduri interne, frunze, rădăcină, înălțime, adâncime
- relații între noduri (părinte, fiu, frate)
- număr minim / maxim de chei pentru înălțimea  $h$
- înălțime minimă / maximă pentru  $n$  noduri
- parcurgeri (RSD, RSD, SRD, niveluri)

## Arbori binari

- 2 Scrieți un algoritm iterativ în pseudo-cod care are ca parametru un arbore binar și care returnează frunzele arborelui în ordine sortată a cheilor.

## Arbori binari

- ③ Considerând un arbore binar, pentru care se cunosc parcurgerile:  
**RSD** (preordine): 10, 7, 6, 15, 22, 11, 45, 16, 43, 9, 12  
**SRD** (inordine): 6, 15, 7, 22, 10, 16, 45, 43, 11, 12, 9  
să se refacă arborele binar.

# Heap binar max

## Conținuturi teoretice

- arbore binar complet
- proprietățile unui heap
- număr minim / maxim de noduri într-un heap de înălțime  $h$
- construirea unui heap max / min dintr-un vector (construct heap)
- inserarea unui nod într-un heap min / max
- extragerea maximului (heap max) / minimului (heap min)

## Heap binar max

- 4 Să se verifice dacă vectorul  $A = \{29, 10, 13, 17, 19, 5, 16, 2, 18, 11\}$  are structura de heap-max. Dacă nu, să se construiască un heap max cu algoritmul corespunzător și apoi să se extragă maximum.  
Cum a arăta în final heap-ul?



## Heap binar min

- 5 Să consideră un heap binar min. Cum poate fi determinată valoarea maximă și ce complexitate are un astfel de algoritm?

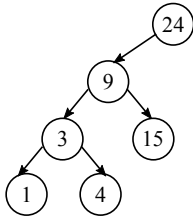
# Arbori binari de căutare

## Conținuturi teoretice

- Proprietățile unui arbore binar de căutare
- Inserție, ștergere a unei chei dintr-un arbore binar de căutare
- Succesor, predecesor al unui nod.

## Arbori binari de căutare

- 6 Se consideră următorul arbore binar de căutare construit prin inserții succesive de chei:



Care dintre următoarele afirmații sunt adevărate?

- a. Cheia 15 a fost inserată sigur după cheia 4
- b. Nu se poate spune care dintre cheile 3 și 4 a fost inserată mai întâi
- c. Arborele din figură este AVL.
- d. Arborele din figură nu este AVL, dar e suficientă o operație de rotație pentru a-l transforma într-un AVL.
- e. În parcurgerea în postordine a arborelui, nodul cu cheia 15 va fi parcurs după nodul cu cheia 9

# Arbori binari de căutare echilibrați

## Conținuturi teoretice

- Operația de rotație într-un arbore binar de căutare
- **Arbore AVL:**
  - Factor de balansare al unui nod
  - Proprietăți ale unui AVL
  - Inserția, ștergerea unei chei dintr-un AVL
- **Arbore roșu - negru (ARN):**
  - Proprietăți ale unui ARN
  - Înălțime neagră
  - Înălțime minimă / maximă pentru o înălțime neagră dată
  - Număr minim/maxim de noduri roșii (sau negre) pentru o înălțime neagră dată
  - Inserția într-un ARN

## Arbori binari de căutare echilibrați

- 7 Se consideră un arbore echilibrat  $T$  (ARN sau AVL). Scrieți o funcție în pseudo-cod, care are ca parametru arborele  $T$  și o valoare  $k$  și care returnează numărul de chei din arbore mai mici decât  $k$ . Ce complexitate are algoritmul dvs? Cum putem rezolva această problemă în complexitate logaritmică? Dacă folosiți îmbogățirea, atunci precizați, cu ce câmp (câmpuri) îmbogățiti și care este relația de recurență folosită pentru a satisface teorema îmbogățirii.

## Arbori AVL

- 8 Se consideră un arbore AVL construit prin inserția a  $n$  noduri. Depinde arborele rezultat de ordinea de inserție a nodurilor sau nu? Justificați.

## Arbori Roșu - Negru

- 9 Dați exemplu de un ARN cu înălțimea neagră  $bh = 3$ , cu un singur nod roșu și cheile mai mari decât 20. Insearați apoi în acesta următoarele chei: 10, 7, 9, 14, 5, 3.

# B-Arbori

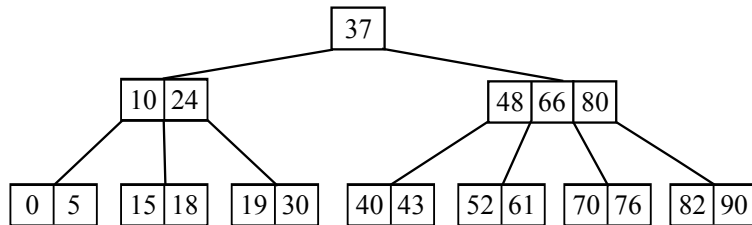
## Conținuturi teoretice

- Proprietățile unui B-arbore
- Grad minim
- Înălțime minimă maximă pentru un grad minim dat și un număr de chei date
- Numă minim/maxim de chei pentru un grad minim dat și o înălțime  $h$  dată
- Cereri într-un B-arbore
- Inserție într-un B-arbore



# B-Arbori

- 10 Se consideră B-arborele din figură cu  $t = 2$ .



Să se insereze pe rând cheile 7,12,44,35,22,9.

# Tabele de repartizare

## Conținuturi teoretice

- Tabele de repartizare cu liste înlănțuite. Inserare în tabelă
- Tabele de repartizare cu dublă repartizare. Inserare în tabelă
- Factor de încărcare.

## Tabele de repartizare

- 11 Considerând o tabelă de repartizare cu liste înlănțuite și cu o funcție de repartizare de tipul  $h(k) = k \bmod m$ ,  $m = 11$ , să se insereze în listă cheile 30, 15, 19, 52, 37, 25, 73, 62. Care este factorul de încărcare al tablei?

## Tabele de repartizare

- 12 Inserați următoarele chei: 30, 15, 19, 52, 37, 25, 73, 62 într-o tabelă de repartizare de dimensiune  $m = 11$  utilizând repartizarea deschisă cu dublă repartizare, cu funcțiile de repartizare auxiliare  $h_1(k) = k \bmod m$  și  $h_2(k) = 1 + k \bmod (m - 1)$ .

# Biblioteca STL

Tipuri de subiecte:

- Pentru o anumită secvență de cod, să se specifice rezultatul (ce va afișa codul)
- Să se cunoască modul de funcționare (stive, cozi, liste, priority\_queue, map, unordered\_map)
- Iteratori
- Comparație complexitate pentru anumite operații
- Ce structură, în ce situație se aplică.