

# 1 Tema Nr. 7: Statistici dinamice de ordine

**Timp alocat:** 2 ore

## 1.1 Implementare

Se cere implementarea **corectă** și **eficientă** a operațiilor de management ale unui **arbore de statistică de ordine** (*capitolul 14 din carte*<sup>1</sup>).

Se cere să folosiți un *arbore binar de căutare* **perfect echilibrat**. Fiecare nod din arbore trebuie extins cu un câmp *size* (dimensiunea sub-arborelui ce are nodul ca rădăcină).

Operațiile de management ale unui **arbore de statistică de ordine**:

- BUILD\_TREE(*n*)
  - *construiește* un arbore binar de căutare **echilibrat** cu cheile 1,2,...,n (*hint: divide et impera*)
  - nu uitați să inițializați câmpul *size*
- OS\_SELECT(*tree*, *i*)
  - selectează elementul cu a *i*-a cea mai mică cheie
  - pseudocodul poate fi găsit la *Capitolul 14.1 din carte(?)*
- OS\_DELETE(*tree*, *i*)
  - puteți folosi ștergerea dintr-un arbore binar de căutare, fără a crește înălțimea arborelui (De ce nu trebuie să re-balansați arborele?)
  - nu uitați să păstrați câmpul *size* consistent o dată cu ștergerile din arbore
  - există mai multe abordări prin care puteți modifica câmpul *size* fără a crește complexitatea algoritmului (găsiți cea mai bună soluție)

Seamănă OS\_SELECT cu ceva ce ați studiat în acest semestru?

## 1.2 Cerințe

### 1.2.1 BUILD\_TREE: implementare corectă și eficientă (5p)

Corectitudinea algoritmilor va trebui demonstrată pe date de intrare de dimensiuni mici (11)

- afișați (cu pretty print) arborele construit inițial

### 1.2.2 OS\_SELECT: implementare corectă și eficientă (1p)

Corectitudinea algoritmilor va trebui demonstrată pe date de intrare de dimensiuni mici (11)

- executați OS-SELECT pentru câțiva (cel puțin 3) indecși selectați aleator.

### 1.2.3 OS\_DELETE: implementare corectă și eficientă (2p)

Corectitudinea algoritmilor va trebui demonstrată pe date de intrare de dimensiuni mici (11)

- executați secvența OS-SELECT urmat de OS-DELETE pentru câțiva (cel puțin 3) indecși selectați aleator (3) și *afișați arborele după fiecare execuție*.

### 1.2.4 Evaluarea operațiilor de management - BUILD, SELECT, DELETE (2p)

! Înainte de a începe să lucrați pe partea de evaluare, asigurați-vă că aveți un **algoritm corect**!

După ce sunteți siguri că algoritmul funcționează corect:

- variați  $n$  de la 100 la 10000 cu un pas de 100;
- pentru fiecare  $n$  (nu uitați să repetați de 5 ori)
  - construiți (BUILD) arborele cu elemente de la 1 la  $n$
  - repetați de  $n$  ori secvența OS-SELECT urmat OS-DELETE folosind un index selectat aleator dintre elementele rămase în arbore
  - Evaluați numărul de operații necesare pentru fiecare operație de management (BUILD, SELECT, DELETE – *reprezentați rezultatele sub forma unui grafic cu trei serii*). Evaluați complexitatea operațiilor de management ca și suma atribuirilor și a comparațiilor pentru fiecare valoare a lui  $n$ .

### 1.2.5 Bonus: Implementarea utilizând AVL / arbori roșu și negru (1p)