

# 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 [1]*).

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 [1]*
- **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 minimale pentru notare

Lipsa oricărei cerințe minimale (chiar și parțială) poate rezulta într-o notă mai mică prin penalizări sau refuzul de a prelua tema, rezultând în nota 0.

- *Demo:* Pregătiți un exemplu pentru exemplificarea corectitudinii fiecărui algoritm implementat. Corectitudinea fiecărui algoritm se demonstrează printr-un exemplu simplu (maxim 10 valori).
- Graficele create trebuie să fie ușor de evaluat, adică grupate și adunate prin funcțiile Profiler după cerințele temei. Tema nu va fi evaluată dacă conține o multitudine de grafice negrupate. De exemplu, analiza comparativă implică gruparea într-un singur grafic a algoritmilor comparați.

- Interpretați graficul/graficele și notați observațiile personale în antetul fișierului *main.cpp*, într-un comentariu bloc informativ.
- Nu preluăm teme care nu sunt indentate și care nu sunt organizate în funcții (de exemplu, nu prelăum teme unde tot codul este pus în main).
- *Punctajele din barem sunt corespondente unei rezolvări corecte și complete a cerintei, calitatea interpretărilor din comentariul bloc și răspunsul corect dat de dumea vostră la întrebările puse de către profesor.*

### 1.3 Cerințe

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

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

- afișați (cu pretty print) arborele construit inițial

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

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

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

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

*Demo:* 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) indecsi selectați aleator (3) și afișați arborele după fiecare execuție.

#### 1.3.4 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 rămasse în arbore

- Evaluati numărul de operații necesare pentru fiecare operație de management (BUILD, SELECT, DELETE – *reprezentați rezultatele sub forma unui grafic cu trei serii*). Evaluati complexitatea operațiilor de management ca și suma atribuirilor și a comparațiilor pentru fiecare valoare a lui  $n$ .

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

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

## **References**

- [1] Thomas H. Cormen et al. *Introduction to Algorithms*. 2nd. The MIT Press, 2001. ISBN: 0262032937.