

# Tema nr. 1: Analiza și Compararea Metodelor Directe de Sortare

**Timp alocat:** 2 ore

## Implementare

Se cere implementarea **corecta** și **eficienta** a 3 metode directe de sortare (sortarea bulelor, sortarea prin inserție – folosind inserție liniară sau binară, și sortarea prin selecție)

Intrare: un șir de numere  $\langle a_1, a_2, \dots, a_n \rangle$

Ieșire: o permutare ordonată a șirului de la intrare  $\langle a'_1 \leq a'_2 \leq \dots \leq a'_n \rangle$

Toate informațiile necesare și pseudo-codul se găsesc în notițele de la seminarul nr. 1 (sortarea prin inserție este prezentată și în carte<sup>1</sup> – secțiunea 2.1). Pentru fiecare metodă de sortare ar trebui să selectați versiunea eficientă (atunci când au fost prezentate mai multe versiuni).

## Praguri de notare

Prag	Cerințe
5	Implementarea unei metode directe de sortare, exemplificare corectitudine și evaluare (cel puțin cazul mediu statistic) - cel puțin 1 grafic
7	Compararea a două metode directe de sortare (în cazurile favorabil, defavorabil și mediu statistic) - implementare, exemplificare corectitudine și analiza (grafice)
9	Compararea a trei metode directe de sortare (în cazurile favorabil, defavorabil și mediu statistic) - implementare, exemplificare corectitudine și analiza (grafice)
10	Discuție, interpretare, eficiență, comparații, stabilitate

## Evaluare

**!** Înainte de a începe să lucrați pe partea de evaluare a complexității algoritmilor, asigurați-vă că aveți un algoritm corect! Corectitudinea algoritmilor va trebui demonstrată pe un vector de dimensiuni mici (care poate să fie codat în funcția „main”).

---

<sup>1</sup> Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. *Introduction to Algorithms*

1. Se cere compararea celor 3 algoritmi în cazurile: **favorabil (best)**, **mediu statistic (average)** și **defavorabil (worst)**. Pentru cazul **mediu** va trebui să repetați măsurătorile de **m** ori ( $m=5$  este suficient) și să raportați media rezultatelor; de asemenea, pentru cazul **mediu**, asigurați-vă că folosiți **aceleași** date de intrare pentru cele 3 metode de sortare (astfel încât compararea lor să fie corectă); identificați și generați date de intrare pentru cazurile: **favorabil** și **defavorabil**, pentru toate cele 3 metode de sortare.
2. Pașii de analiză ai metodelor de sortare pentru fiecare din cele 3 cazuri (**favorabil**, **defavorabil**, **mediu**):
  - variați dimensiunea șirului de la intrare ( $n$ ) între  $[100 \dots 10.000]$ , cu un increment de maxim 500 (sugerăm 100);
  - pentru fiecare dimensiune, generați datele de intrare adecvate pentru metoda de sortare; rulați metoda de sortare numărând operațiile (numărul de atribuiri, numărul de comparații și suma lor).

! Doar atribuiri („=”) și comparațiile („<”, „==”, „>”, „!=”) care se fac pe datele de intrare și pe datele auxiliare corespunzătoare se iau în considerare.
3. Pentru fiecare caz de analiză (**favorabil**, **defavorabil** și **mediu**), generați grafice care compara cele 3 metode de sortare; folosiți grafice diferite pentru numărul de atribuiri, comparații și suma lor. Dacă o curbă nu poate fi vizualizată corect din cauza că celelalte curbe au o rată mai mare de creștere (ex: o funcție liniară pare constantă atunci când este plasată în același grafic cu o funcție pătratică), atunci plasați noua curbă și pe un alt grafic.  
Denumiți adecvat graficele și curbele.
4. Interpretați graficele și scrieți observațiile în antetul fișierului „.cpp”, într-un comentariu bloc.