

# Tema Nr. 4: Interclasarea eficientă a k şiruri ordonate

Timp alocat: 2 ore

## Implementare

Se cere implementarea **corectă** şi **eficientă** a unei metode de complexitate  $(n * \log k)$  pentru **interclasarea a k şiruri sortate**. Unde n este numărul total de elemente (Sugestie: folosiţi un heap, vezi notiţele de la seminarul al 2-lea).

Cerinţe de implementare:

- Folosiţi liste înlănţuite pentru a reprezenta cele k şiruri sortate şi secvenţa de ieşire

Intrare: k şiruri de numere  $\langle a_1^i, a_2^i, \dots, a_{m_i}^i \rangle$ ,  $\sum_{i=1}^k m_i = n$

Ieşire: o permutare a reuniunii şirurilor de la intrare  $a_1' \leq a_2' \leq \dots \leq a_n'$

## Praguri notare

Nota	Cerinţe
5	Generare k liste aleatoare sortate de dimensiuni diferite, având în total n elemente (n si k date); interclasarea a 2 liste
6	Adaptare operaţiei heap pe structura noua (cheie, index_lista); folosire min-HEAP
8	Implementare corecta si completa al algoritmului de interclasare cu exemplificarea corectitudinii (demo)
10	Evaluarea algoritmului în cazul mediu statistic cu varierea atat k cat si n, interpretare, discuţie

## Evaluare

! Înainte de a începe să lucraţi pe partea de evaluare, asiguraţi-vă că aveţi un **algoritm corect**! Corectitudinea algoritmilor va trebui demonstrată pe date de intrare de dimensiuni mici (ex: k=4, n=20).

Se cere analiza algoritmului în cazul **mediu statistic**. Pentru cazul **mediu statistic** va trebui să repetaţi măsurătorile de câteva ori. Din moment ce **k** şi **n** pot varia, se va face o analiză în felul următor:

1. Se alege, pe rând, 3 valori constante pentru **k** ( $k_1=5$ ,  $k_2=10$ ,  $k_3=100$ ); generează **k** șiruri **aleatoare** sortate pentru fiecare valoare a lui **k** astfel încât numărul elementelor ( $m_k$ ) din toate șirurile să varieze între 100 și 10000 cu un increment maxim de 400 (sugerăm 100); rulați algoritmul pentru toate valorile lui **n** (pentru fiecare valoare a lui **k**); generați un grafic ce reprezintă suma atribuirilor și a comparațiilor făcute de acest algoritm pentru fiecare valoare a lui **k** (în total sunt 3 curbe).
2. Se alege **n=10.000**; valoarea lui **k** va varia între 10 și 500 cu un increment de 10; generați **k** șiruri **aleatoare** sortate pentru fiecare valoare a lui **k** astfel încât suma lungimii fiecărui șir să fie 10000; testați algoritmul de interclasare pentru fiecare valoare a lui **k** și generați un grafic care reprezintă suma atribuirilor și a comparațiilor.
3. Interpretați graficele într-un comentariu bloc informativ.
4. ***Punctajele din barem se dau pentru rezolvarea corectă și completă a cerinței, calitatea interpretărilor din comentariul bloc și răspunsul corect dat de voi la întrebările puse de către profesor***