

Tema Nr. 5: Căutarea în tabele de dispersie

Adresare deschisa, verificare pătratică

Timp alocat: 2 ore

Implementare

Se cere implementarea **corectă** și **eficientă** a operațiilor de *inserare* și *căutare* într-o tabela de dispersie ce folosește adresarea deschisa cu verificare pătratică.

Informații utile și pseudo-cod găsiți în notițele de curs sau în carte, în secțiunea *11.4 Open addressing*.

Notiunile închis/deschis (closed/open) specifica daca se obliga folosirea unei pozitii sau structuri de date.

Hashing (se refera la tabela de dispersie (hash table))

- Open Hashing
 - Pe o anumita pozitie se pot stoca mai multe elemente ex: chaining
- Closed Hashing
 - Se poate stoca doar un singur element pe o anumita pozitie ex.
linear/quadratic probing

Addressing (se refera la pozitia finala a unui element fata de pozitia initiala)

- Open Addressing (Adresare Deschisa)
 - Adresa finala (pozitia finala) nu este complet determinat de catre codul hash. Pozitia depinde si de elementele care sunt deja in tabela. ex: linear/quadratic probing (verificare liniara/patratica)
- Closed Addressing (Adresare Inchisa)
 - Adresa finala este intotdeauna determinata the codul hash (pozitia initiala calculata) si nu exista probing (verificare) ex: chaining

Praguri notare

Nota	Cerințe
5	Implementarea operațiilor de inserare și căutare într-o tabela de dispersie; demo pe factor de umplere 95%
7	Evaluarea operației de căutare
9	Implementare corectă și completă a algoritmului, cu demo
10	Evaluare, interpretare, discuție

Evaluare

! Înainte de a începe să lucrați pe partea de evaluare, asigurați-vă că aveți un **algoritm corect**! Corectitudinea algoritmilor va trebui demonstrată pe date de intrare de dimensiuni mici.

Se cere evaluarea operației de *cautare* în tabele de dispersie cu adresare deschisă și verificare pătratică, în cazul **mediu statistic** (nu uitați să repetați măsurătorile de 5 ori). Pentru a obține evaluarea, trebuie să:

1. Alegeți N , dimensiunea tabelului, un număr prim în jur de 10000 (e.g. 9973, sau 10007);
2. Pentru fiecare din următoarele valori pentru factorul de umplere $\alpha \in \{0.8, 0.85, 0.9, 0.95, 0.99\}$:
 - a. Inserați în tabela n elemente aleator, astfel încât să ajungeți la valoarea lui α ($\alpha = n/N$)
 - b. Căutați, în fiecare caz, m elemente aleator ($m \sim 3000$), astfel încât aproximativ jumătate din elemente să fie *gasite*, iar restul să *nu fie gasite* (în tabela). Asigurați-vă că elementele *gasite* sunt generate uniform, i.e. să căutați elemente care au fost introduse la momente diferite, cu probabilitate egală (există mai multe moduri în care se poate asigura acest lucru)
 - c. Numărați operațiile efectuate de procedura de cautare (i.e. numărul de celule accesate)

3. Generati un tabel de forma:

Factor de umplere	Efort mediu <i>gasite</i>	Efort maxim <i>gasite</i>	Efort mediu <i>ne-gasite</i>	Efort maxim <i>ne-gasite</i>
0.8				
0.85				
...	

Efort mediu = efort_total / nr_elemente

Efort maxim = numar maxim de acese efectuat de o operatie de cautare

4. Interpretati rezultatele.