Recapitulare Structuri de Date (Liste, Skip Lists, Hashing)

1. Liste legate

- Tipuri:
- Simplu legata: fiecare nod are valoare + pointer la urmatorul.
- Dublu legata: fiecare nod are valoare + prev + next.
- Circulara: ultimul nod pointeaza la primul.
- Operatii si complexitate:
- List-Insert: O(1)
- List-Delete (cu pointer la nod): O(1)
- List-Search: O(n)
- Avantaje: inserare/stergeri rapide.
- Dezavantaje: acces lent la un element arbitrar (trebuie parcursa lista).

2. Skip Lists

- Extensie a listelor ordonate: nodurile au pointeri suplimentari pe mai multe niveluri.
- Permite salturi mari, deci cautare mai rapida.
- Complexitate medie:
- Search: O(log n)
- Insert: O(log n)
- Delete: O(log n)
- Complexitate worst-case: O(n) (rar).
- Aplicatii: baze de date (Redis, Lucene), retele, motoare de cautare.

3. Hash Tables

Idee

- Functie hash h(k) care mapeaza cheia k in {0,...,m-1}.
- Operatii dorite: Insert, Delete, Search = O(1) in medie.

Probleme: coliziuni

- Doua chei diferite k1, k2 pot avea h(k1)=h(k2).

Rezolvare coliziuni

1. **Chaining**
- Fiecare bucket contine o lista (de obicei legata).

- Complexitate medie: O(1 + alpha), unde alpha = n/m (load factor).

- Avantaj: usor de implementat, stergeri simple.

2. **Open Addressing**

- Toate cheile sunt stocate direct in tabela.

- Conditie: alpha <= 1.

- Tehnici de probing:

- Linear probing: h(k,i) = (h'(k) + i) mod m (sufera de primary clustering).

- Quadratic probing: h(k,i) = (h'(k) + c1*i + c2*i^2) mod m (secondary clustering).

- Double hashing: h(k,i) = (h1(k) + i*h2(k)) mod m (cea mai buna performanta).

- Stergere: marcam slotul ca DELETED (nu NIL), altfel sirul de probe se rupe.

4. Functii Hash

- Diviziune: h(k) = k mod m (alege m prim, nu putere a lui 2).

- Multiplicare: h(k) = floor(m * frac(k*A)), cu A ~ 0.618 (Knuth).

- Folding: spargi cheia in bucati, le combini si iei mod m.

- Universal hashing: alegem h random dintr-o familie H (Carter-Wegman).

- Perfect hashing: pentru seturi statice, permite Search O(1) worst-case (folosit cand cheile nu se schimba).

5. Bloom Filters

- Structura probabilistica pentru membership test.

- Raspunsuri:

- NO = sigur nu exista.

- YES = posibil, dar poate fi fals pozitiv.

- Avantaj: spatiu redus, rapid.

Intrebari tip examen

1. Care este diferenta intre o lista dublu legata si una simplu legata?

2. De ce Skip Lists ofera Search O(log n) in medie?

3. Care este diferenta intre Chaining si Open Addressing pentru tabele hash?

4. Ce este load factor (alpha) si ce influenta are asupra performantei unei hash table?

5. De ce se foloseste DELETED in loc de NIL la stergere in Open Addressing?

6. Care sunt avantajele si dezavantajele unei functii hash bazata pe diviziune?

7. Ce aplicatii practice pot avea Bloom Filters?