

Recapitulare Arbori (BST, AVL, Red-Black, Splay, B-Trees)

1. Binary Search Tree (BST)

- Proprietate: pentru fiecare nod x , toate cheile din stanga $\leq x.key \leq$ toate cheile din dreapta.
- Operatii:
- Search: $O(h)$
- Insert: $O(h)$
- Delete: $O(h)$
- Problema: daca inserezi elemente sortate, arborele devine degenerat, $h = n$.

2. AVL Tree

- Arbore binar de cautare echilibrat.
- Conditie: $|\text{height}(\text{left}) - \text{height}(\text{right})| \leq 1$.
- Operatii:
- Search: $O(\log n)$
- Insert: $O(\log n) + \text{rotatii}$
- Delete: $O(\log n) + \text{rotatii}$
- Rulaje de rotatii: LL, RR, LR, RL.
- Avantaj: foarte echilibrat (garantat $\log n$). Dezavantaj: mai multe rotatii.

3. Red-Black Tree

- Arbore binar de cautare echilibrat aproximativ, nodurile sunt Rosu/Negru.
- Proprietati:
- 1. Radacina este neagra.
- 2. Frunzele NIL sunt negre.
- 3. Nod rosu nu are copil rosu.
- 4. Toate drumurile simple au acelasi numar de noduri negre.
- Operatii: Search/Insert/Delete: $O(\log n)$.
- Avantaj: mai putine rotatii decat AVL, bun in practica.
- Aplicatii: C++ STL (map, set), Java TreeMap.

4. Splay Tree

- Arbore auto-ajustabil: cand accesezi un nod, il aduci la radacina prin rotatii (splaying).

- Operatii: Search/Insert/Delete: $O(\log n)$ amortizat.
- Avantaj: simplu, excelent pentru acces repetat la aceleasi date.
- Dezavantaj: o operatie poate dura $O(n)$ in cel mai rau caz.

5. B-Trees si B+Trees

- Arbori echilibrati cu multi copii (nu doar 2). Folositi in baze de date si sisteme de fisiere.
- Toate frunzele sunt la acelasi nivel.
- Operatii: Search/Insert/Delete: $O(\log n)$.
- B+Tree: toate valorile sunt doar in frunze, nodurile interne doar pentru chei de ghidaj.

Comparatie arbori

Arbore	Search	Insert	Delete	Echilibru	
-----	-----	-----	-----	-----	
BST simplu	$O(h)$	$O(h)$	$O(h)$	poate fi $h=n$	
AVL	$O(\log n)$	$O(\log n)$	$O(\log n)$	strict echilibrat	
Red-Black	$O(\log n)$	$O(\log n)$	$O(\log n)$	echilibru relaxat	
Splay	$O(\log n)$ amortizat	$O(\log n)$ amortizat	$O(\log n)$ amortizat	auto-ajustare	
B-Tree	$O(\log n)$	$O(\log n)$	$O(\log n)$	echilibrat pe disc	

Intrebari tip examen

1. Care este diferenta intre un BST simplu si un AVL?
2. Cum se face reechilibrarea la AVL si ce tipuri de rotatii exista?
3. Care sunt cele 4 proprietati ale arborilor Red-Black?
4. De ce spunem ca Splay Trees ofera performanta $O(\log n)$ amortizata?
5. Unde sunt folosite in practica B-Trees si de ce?