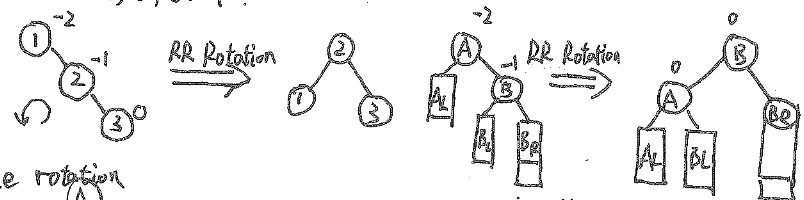
ADS

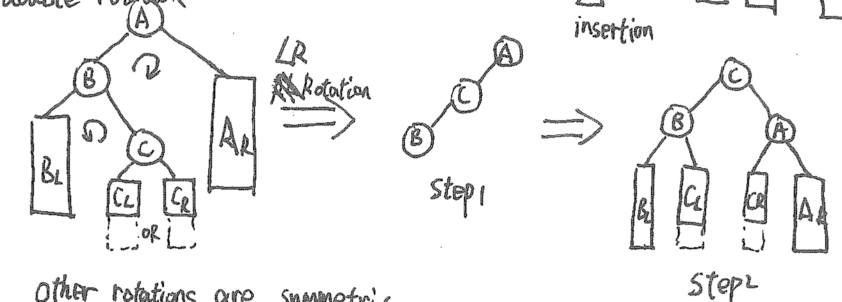
1. AVL 与 Splay 树

The height of an empty tree is defined to be -1.

The balance factor $BF(node) = h_L - h_R$. In an AVL tree, $BF(node) = -1, 0, \text{or } 1$.



double rotation



other rotations are symmetric

n_h be the minimum nuber of nodes in the height balanced tree of height h.

$n_h = n_{h-1} + n_{h-2} + 1$

$n_h = F_{h+2} - 1 \quad F_i \approx \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^i \Rightarrow n_h \approx \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{h+2} - 1 \Rightarrow h = O(\ln n)$

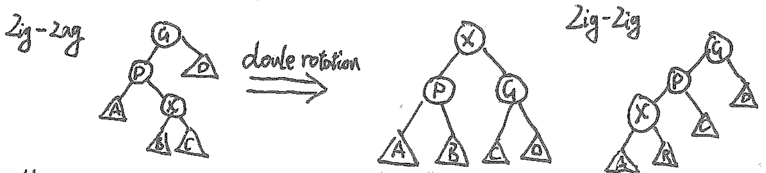
splay tree:

any M consecutive tree operations starting from an empty tree take at most $O(M \log N)$ time.

After a node is accessed, it is pushed to the root by a series of AVL tree rotations.

Case 1: P is the root → Rotate X and P

Case 2: P is not the root



delections:

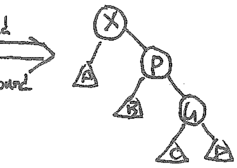
Step 1: Find X;

Step 2: Remove X;

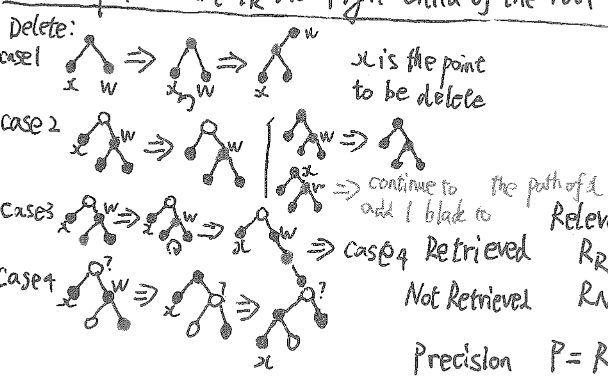
Step 3: FindMax(T_L);

Step 4: Make T_R the right child of the root of T_L

worst-case bound ≥ amortized bound ≥ average-case bound



Delete:



2. Red-Black tree

(1) every node is black or red

(2) root is black

(3) every leaf is black (NIL)

(4) if a node is red, both child are black

☆ (5) all path contain the same number of blacknodes

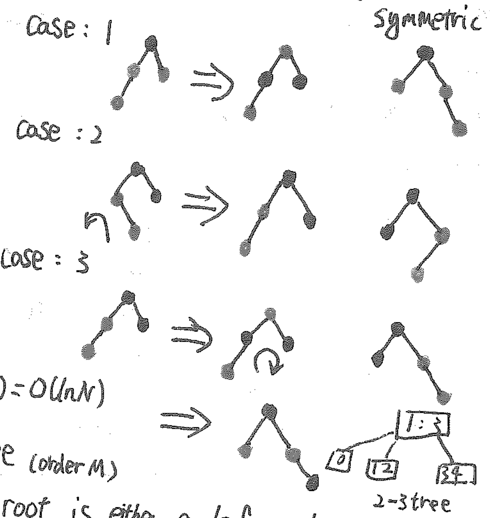
black-height of any node x, denoted by bh(x) is the number of black nodes on any simple path from x (x not included) down to a leaf $bh(Tree) = bh(root)$

☆ N internal nodes at most $2\ln(N+1)$ height

Insert:

color red

-can be down iteratively



$T = O(h) = O(\ln N)$

3. B+ tree (order M)

(1) the root is either a leaf or has 2 and M children

(2) All none nodes (except the root) have between $\lceil M/2 \rceil$ and M children.

(3) All leaves are the same depth

a general B^+ tree of order M

$Depth(M,N) = O(\lceil \log_{\lceil M/2 \rceil} N \rceil)$

$T(M,N) = O((M/\log M)\log N)$

$T_{Find}(M,N) = O(\log N)$

4. Inverted Index FILE

Term - Document Incidence Martix

Compact-Version Inverted File Index

Inverted file contains a list of pointers

NO.	Term	Times; Documents
1	truck	<3; 1,3,4>

$\Rightarrow$ <3; (1;3); (3;8), (4;7)>

```
while ( read a document D ) {
    while ( read a term T in D) {
        if ( Find ( Dictionary, T) == False )
            Insert ( Dictionary, T);
        Get T's posting list;
        Insert a node to T's posting list;
    }
}
write the inverted index to disk.
```

how fast does it index

how fast does it search

~~experi~~ expressiveness of query language

word stremming stop words

access a term: 1. Search trees Hashing

	Relevant	Irrelevant
Retrieved	R_R	I_R
Not Retrieved	R_N	I_N

Precision $P = R_R / (R_R + I_R)$

Recall $R = R_R / (R_R + R_N)$

	Relevant	Irrelevant
Retrived	R_R	I_R
Not Retrived	R_N	I_N

Precision $= \frac{R_R}{R_R + I_R}$ Recall $R = \frac{R_R}{R_R + R_N}$