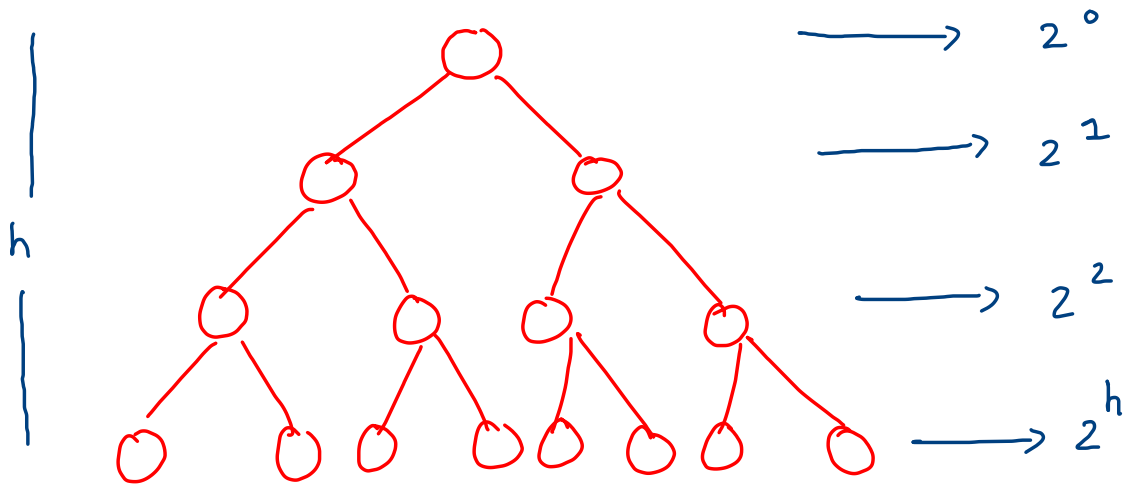


$$n = 2^0 + 2^1 + \cdots 2^h$$

$$n = 1(2^{h+1} - 1)$$

$$n = 2^{h+1} - 1$$

$$h \propto \log_2 n$$

$a = 1$

$r = 2$

$k = h$

$$s = \frac{a(r^k - 1)}{r - 1}$$

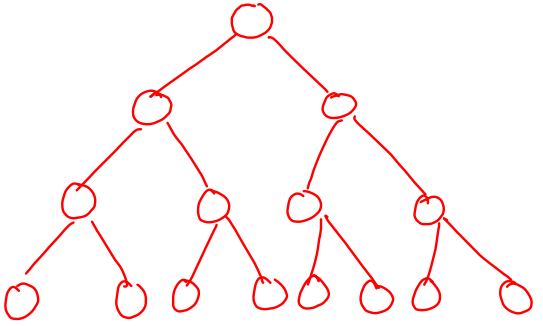
Binary tree

```
//self check
if(node.data == data) {
    return true;
}

//left child
boolean fil = find(node.left,data);

if(fil == true) {
    return true;
}

//right child
boolean fir = find(node.right,data);

if(fir == true) {
    return true;
}

return false;
```

$T \propto O(n)$

BST

```
if(data > node.data) {
    return find(node.right,data);
}
else if(data <  node.data) {
    return find(node.left,data);
}
else {
    return true;
}
```



$T \propto O(h)$

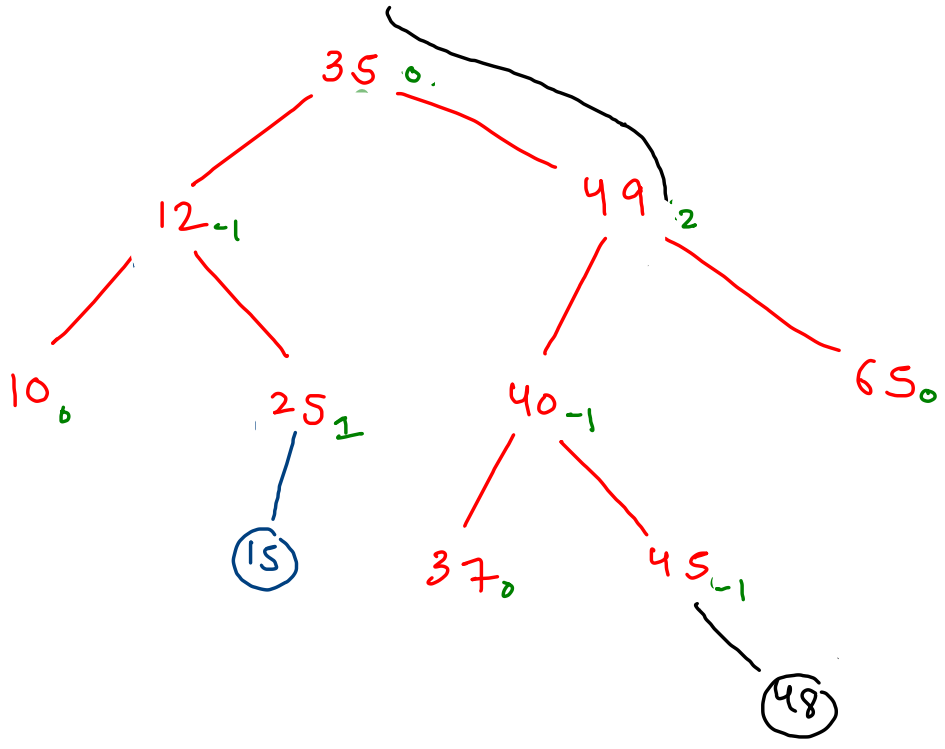
skewed BST

$h \approx n$

AVL

Self balancing BST

add (15) add (48)



bf = lh - rh

|bf| ≤ 1

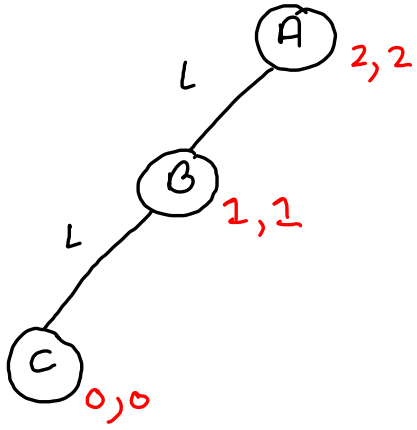
ht -> edges

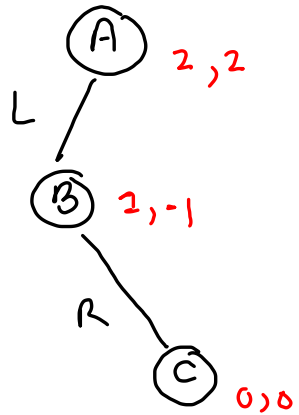
safe bf

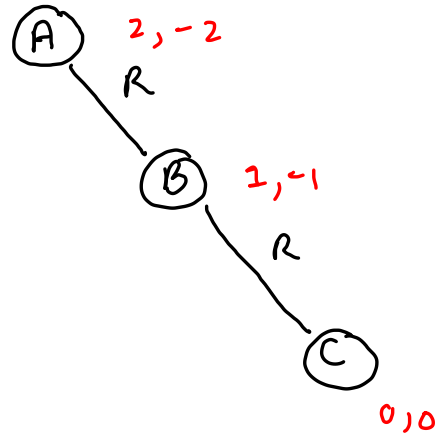
0, 1, -1

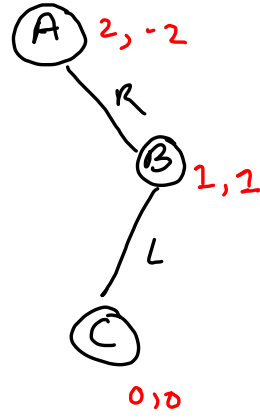
```java
public static Node add(Node node, int data) {
    if(node == null) {
        return new Node(data,null,null);
    }

    if(data > node.data) {
        node.right = add(node.right,data);
    }
    else if(data < node.data) {
        node.left = add(node.left,data);
    }
    else {
        //nothing to do
    }

    return node;
}
```

Problem :







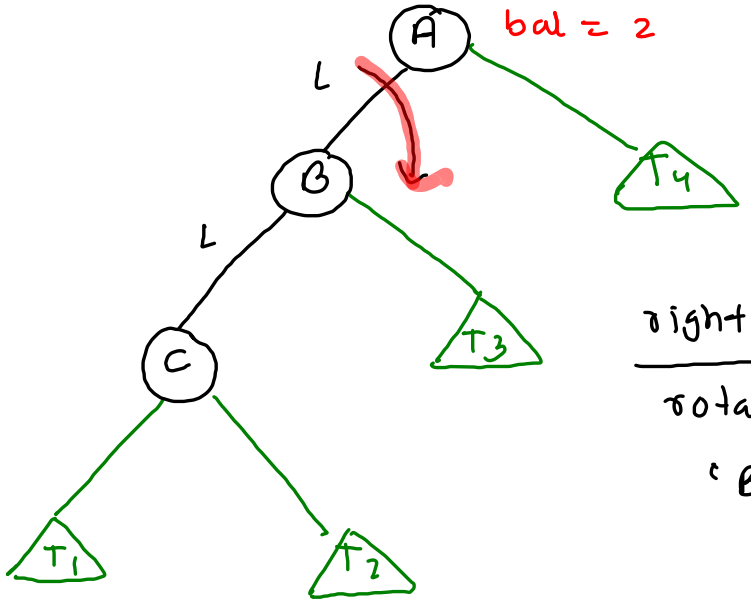


bf = lh - rh

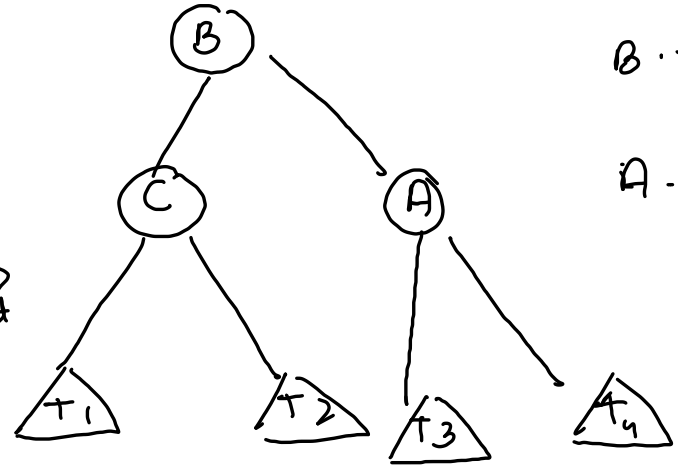
(ht, bf)

Solve Problems

LL

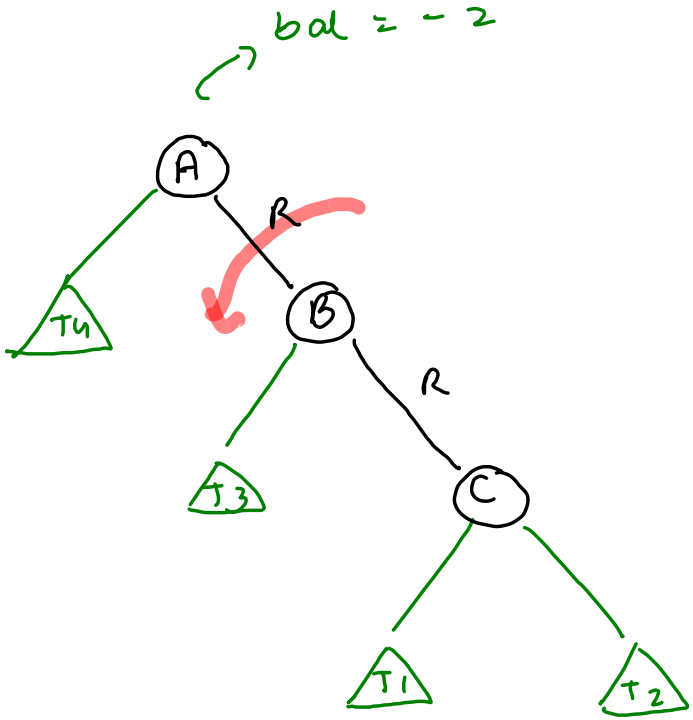


right rotation at 'B'

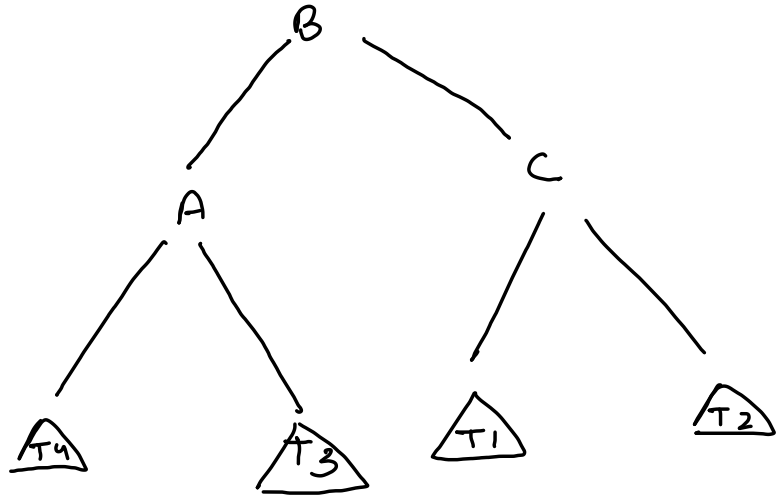



```
B_right = B.right;
B.right = A ;
A.left = B_right;
updateHT Bal(A);
updateHT Bal(B);
return B;
```

RR



left rotation at 'B'



B_left = B.left;

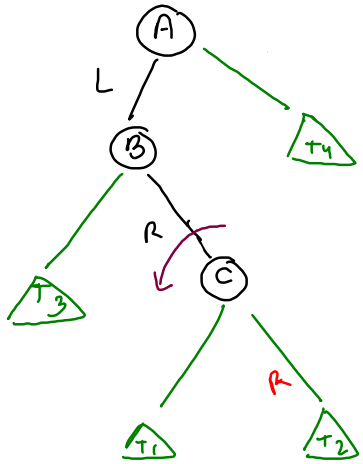
B.left = A;

A.right = B_left;

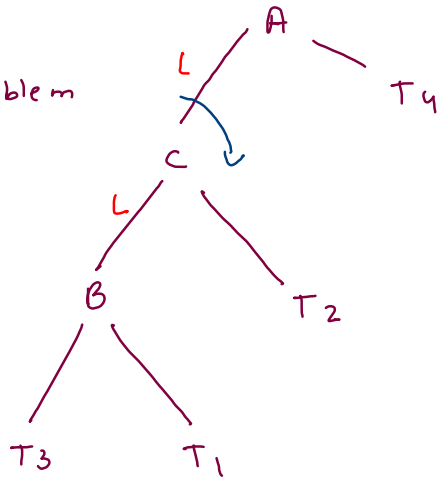
update HT Bal for A, B;

return B;

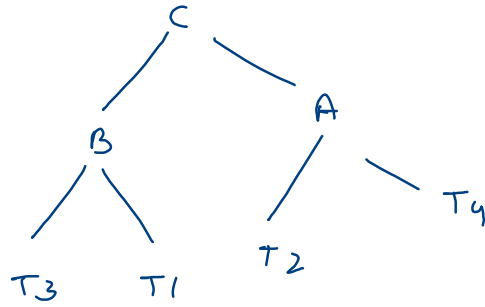
LR



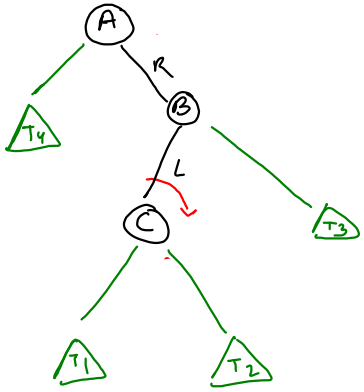
Solve RR problem at B



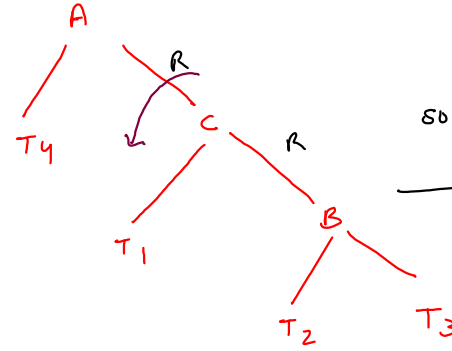
Solve LL problem at A



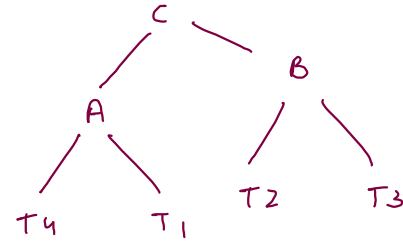
RL

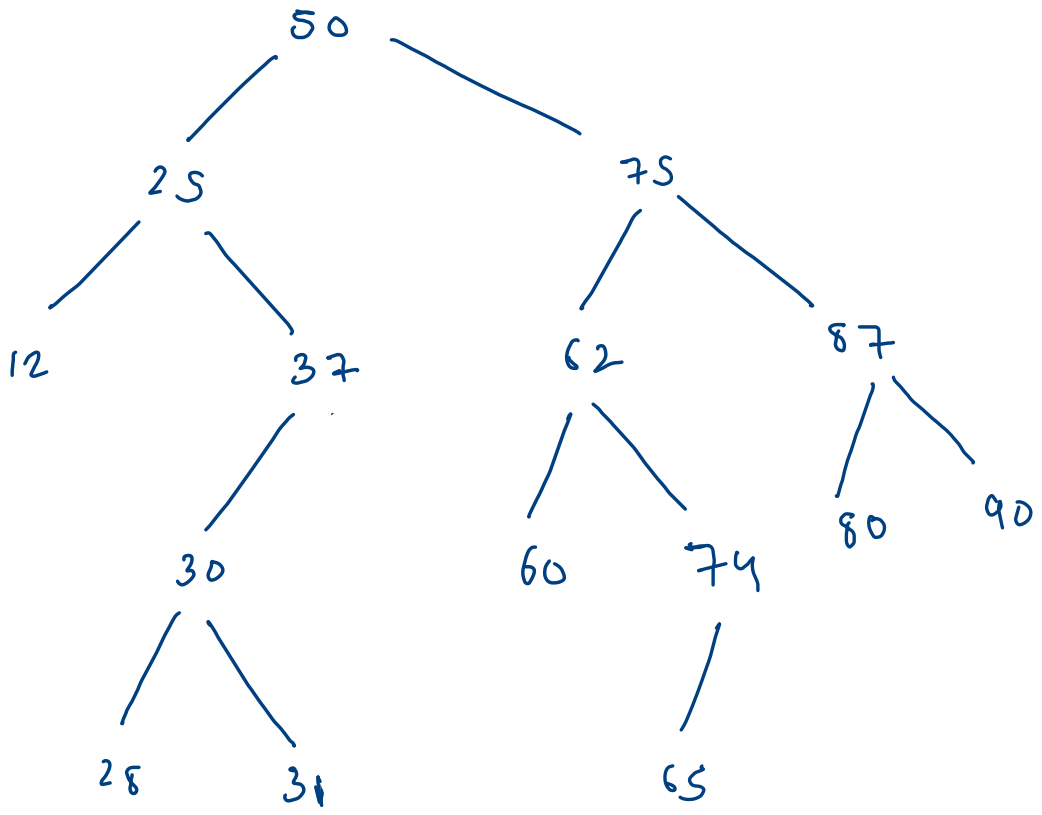


solve LL at 'B'



solve RR at A





28

37

30

75

remove :

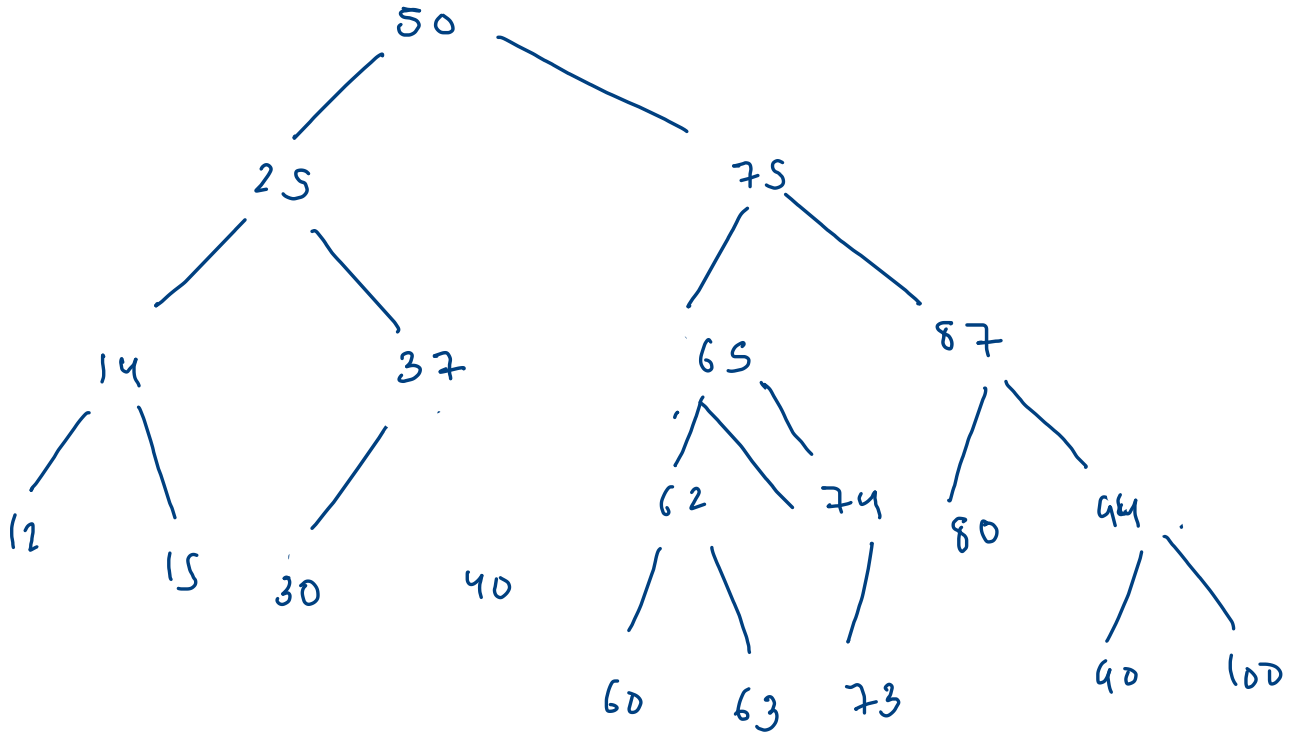
0 child -> return null on my behalf

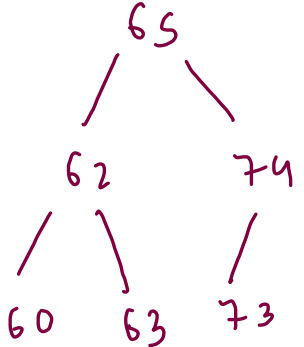
1 child -> return single child on my behalf

2 child :

node.data = dmax

remove(node.left, dmax)





(i) add (15)

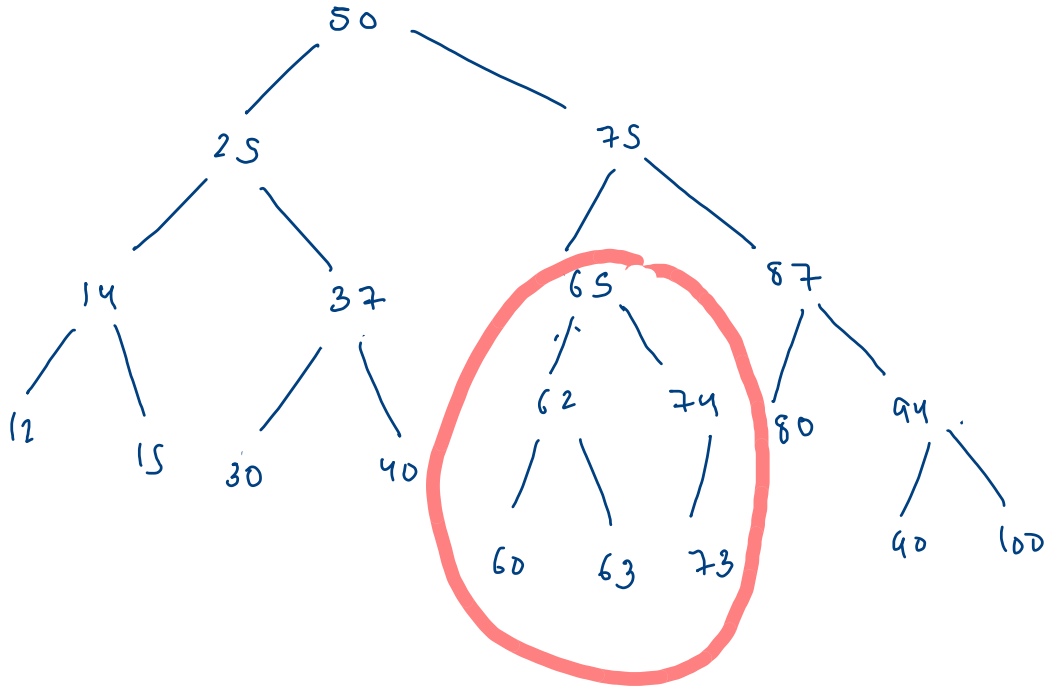
(ii) add (14)

(iii) add (63)

(iv) add (94)

(v) add (100)

(vi) add (73)

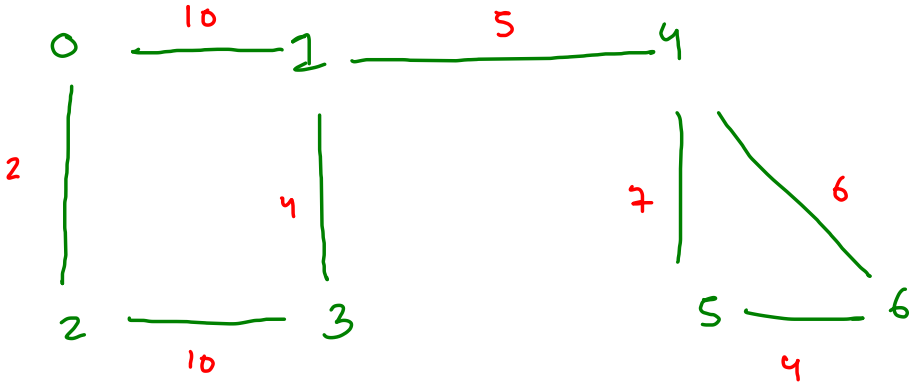

50
25
75
14
37
65
87
12
15
30
40
62
74
80
94
60
63
73
90
100

Graphs

edges and vertices

v = 7

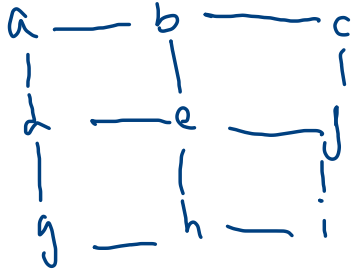
e = 8

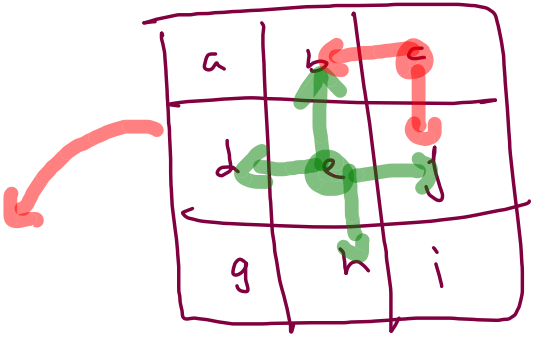


(i) Adjacency list

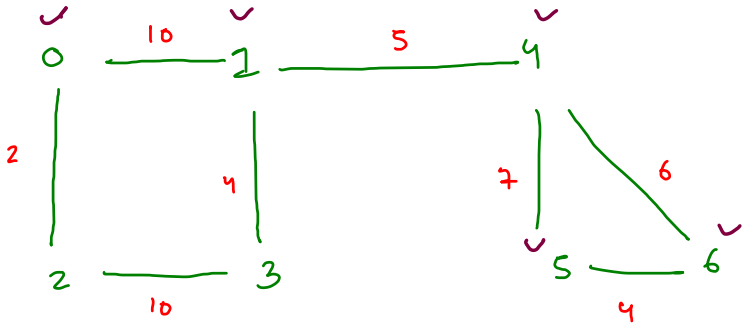
(ii) adjacency matrix

(iii) nbr in matrix



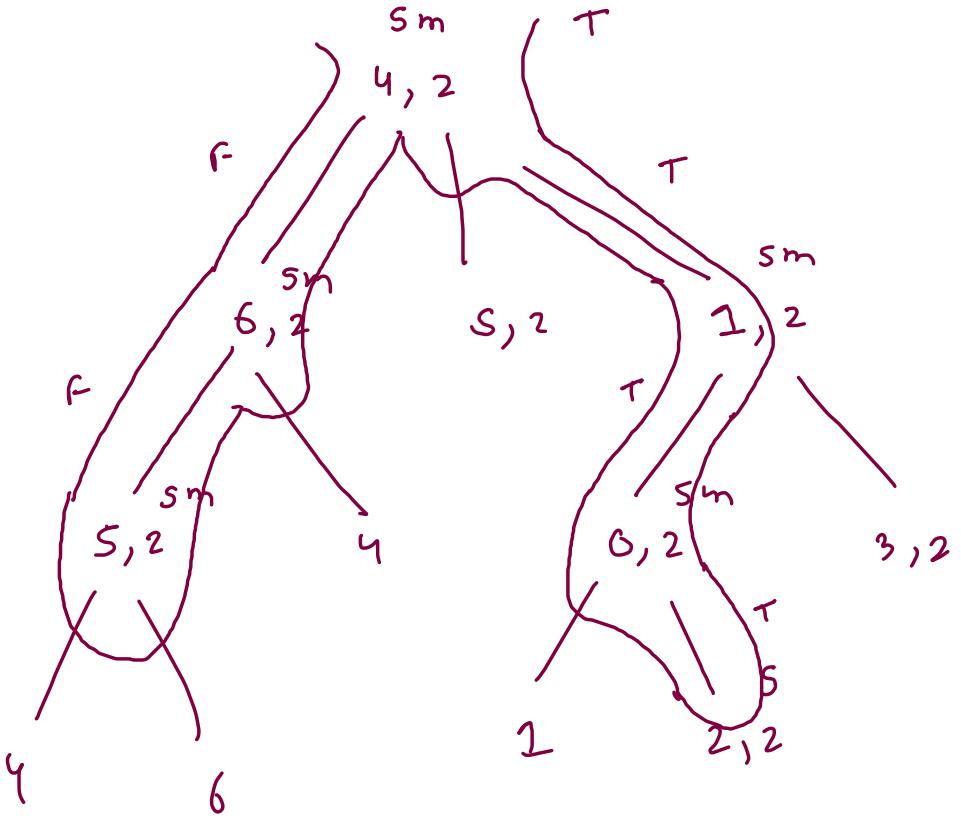


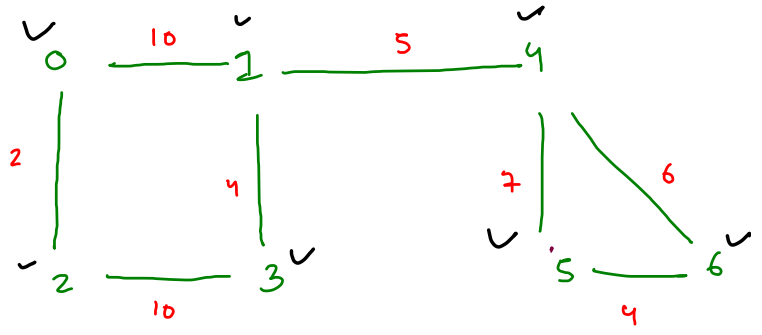
(td dr -> nbr)



hasPath (4 , 2)

src dest





0

1 2

3 4

5 6

count = 0

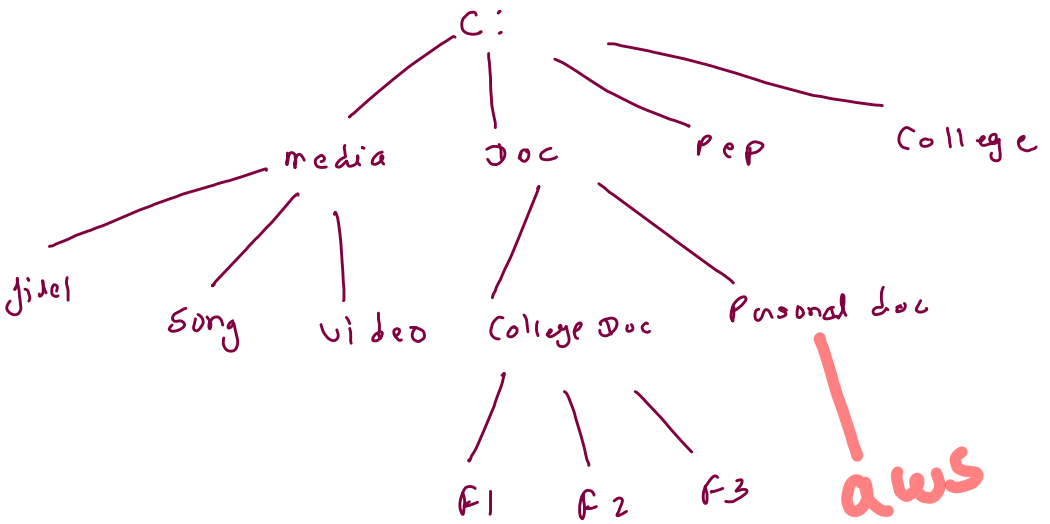
count
- remove
- mark
- work
- add nbr

time

~~0~~	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~	~~5~~	~~6~~	~~6~~

Leetcode : 200,

Lintcode : 860



Commvault

3 DSA
↓
7-8 hour
↓
2 hour

```
Folder {
    String Name;
    HM <S, Folder> cFolder;
    HM <S, File> cFile;
}

File {
    String name;
    String content;
}
```

addFile (c/Doc/Personal_Doc/aws.txt, "Amazon cloud service")

path

content

arr = [C, Doc, Personal_Doc, aws.txt]

aws
Cont