Solution of ST2

Course - B.Tech

Session - 2017-18

Subject - Design and Analysis of Algorithm

Marks - 100

Time - 2 hrs

Semester - V

Sub. Code - NCS-501

Section - IT-1,2, CS-1,2,3

Subject faculties

Mr Kapil Tomar

Mr B. N. Pandey

Ms. Divya Gaple

Ms Gargi

Reviewed 16/10/17

(Ruchin)

Section - A

A- Attempt all the parts.

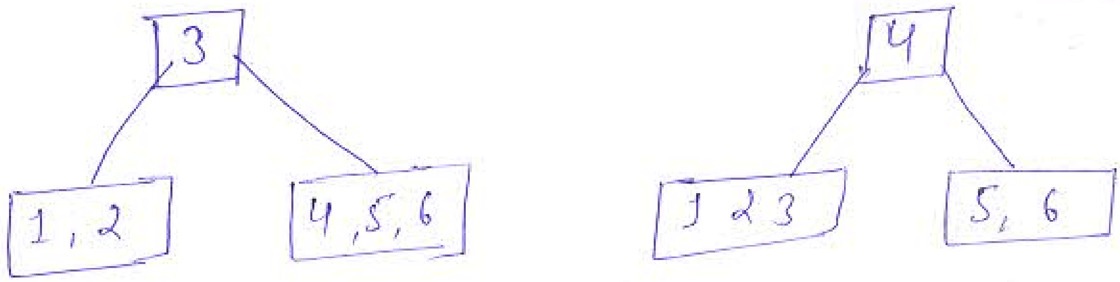
1. What is the largest possible number of internal nodes in red-black tree with black height K?

Ans:- The smallest no of internal nodes in a red black tree with black height of K is 2^{K-1}.

And the largest no of internal nodes in a red black tree with black height of K is $2^{2K}-1$.

2- Show all legal B. tree of t=3 for {1, 2, 3, 4, 5, 6} elements.

Ans:-

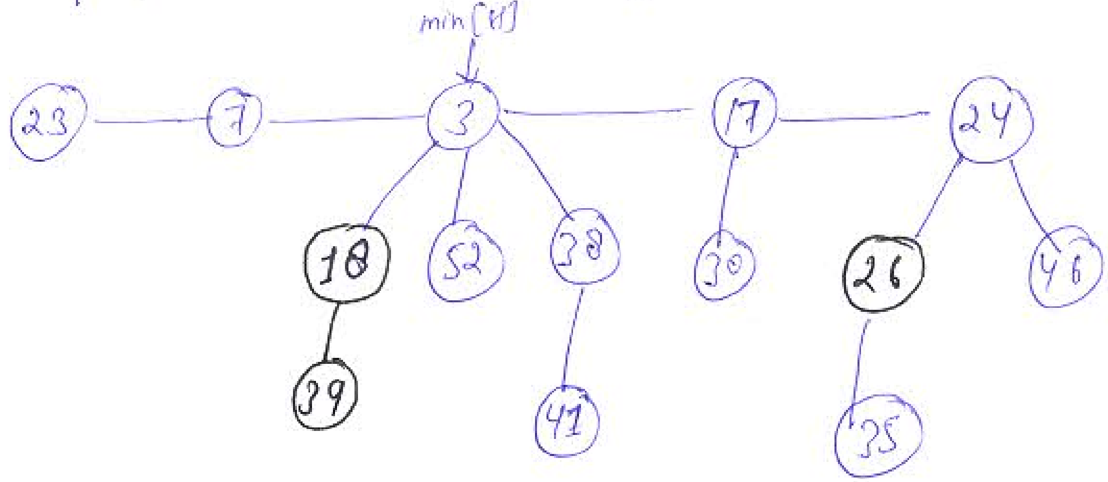


3- What is the potential function of fibbonacci-heap? Give example.

Ans:- we use the potential function to analyze the performace of fibonacci heap operations. For a given Fibonacci heap H, we indicate by t(H) the number of trees in the root list of H and by $m(H)$ the number of marked nodes in H.

The potential of Fibonacci heap H is then defined by -

$$\phi(H) = t(H) + 2m(H)$$



$t(H) = 5$

$m(H) = 3$

$\phi(H) = 5 + 2(3) = 5 + 6 = 11$

4. Give the best case time complexity of binary search in both successful search and unsuccessful search.

Ans. Best case time complexity of binary search in case of successful search is $O(1)$

and best case time complexity of binary search in case of unsuccessful search is $O(\log_2 n)$

5. Explain time complexity of Strassen's Algorithm

Ans: $T(n) = 7\,T(n/2) + O(n^2)$

$a = 7, \quad b = 2 \qquad f(n) = n^2$

$n^{\log_b a} = n^{\log_2 7} = n^{2.81}$

$f(n) \leq n^{\log_b a}$

So $\boxed{T(n) = O(n^{2.81})}$

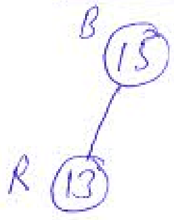
Section - B

6. Insert the nodes 15, 13, 12, 16, 19, 23, 5, 8 in empty Red-Black tree.

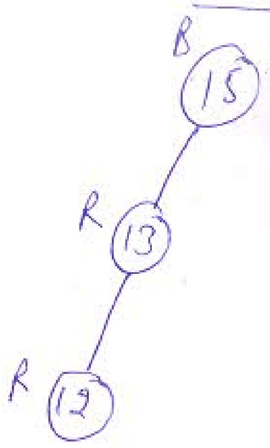
Ans:

insert-15

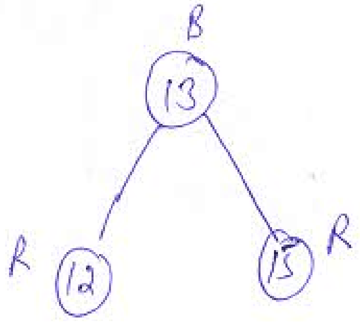
inserting-13



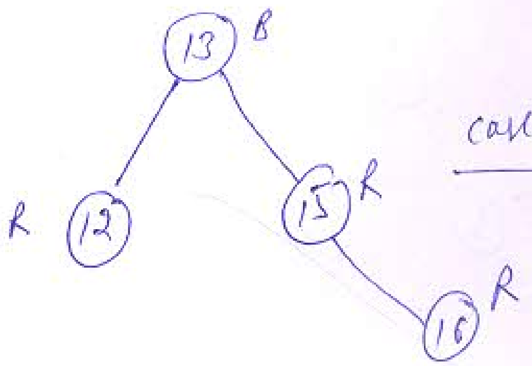
insert-12



case-3 →

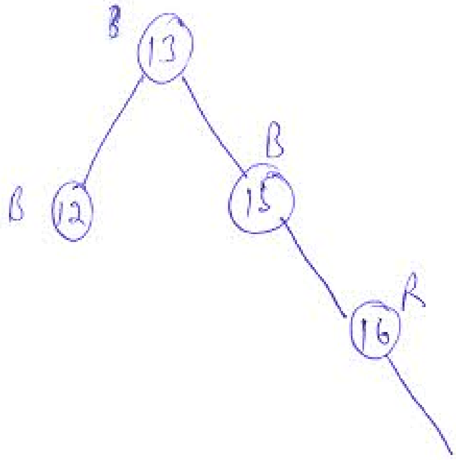


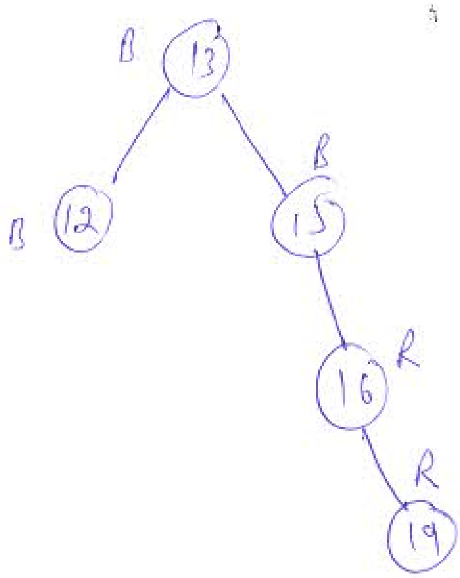
insert-16



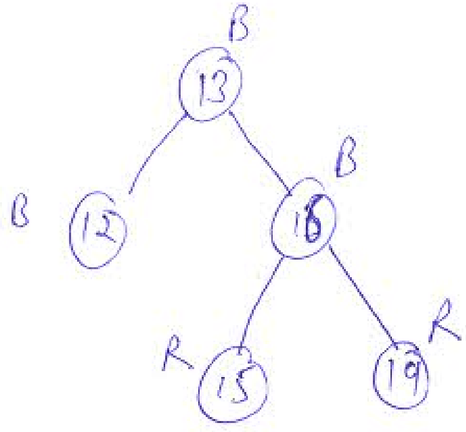
case-1 →

insert-19

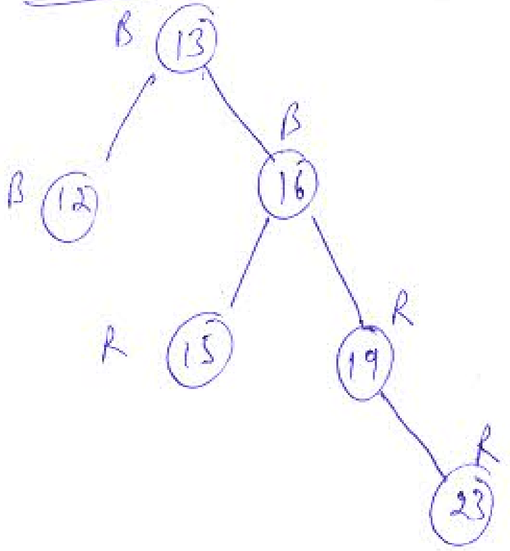




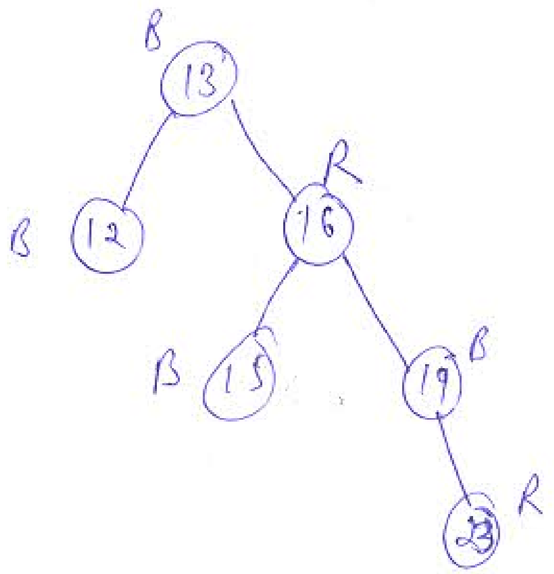
case-3



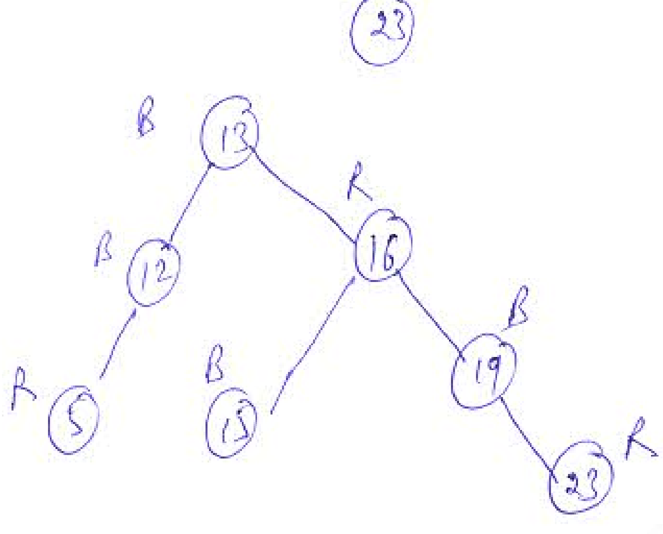
insert 23



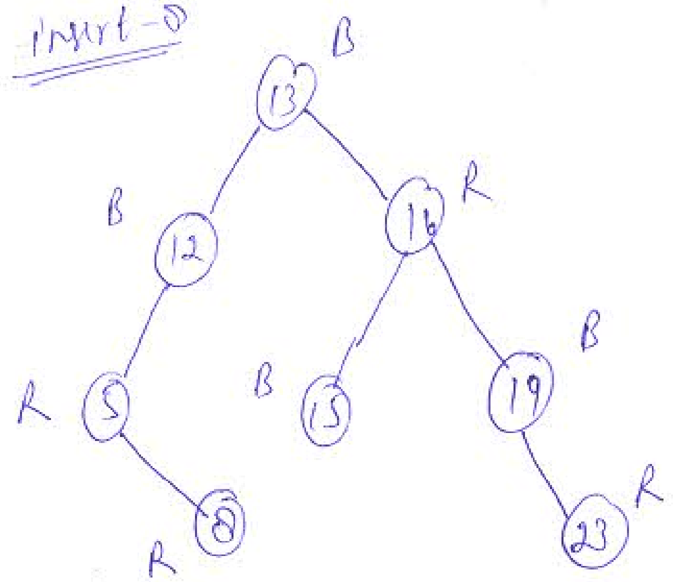
case-1



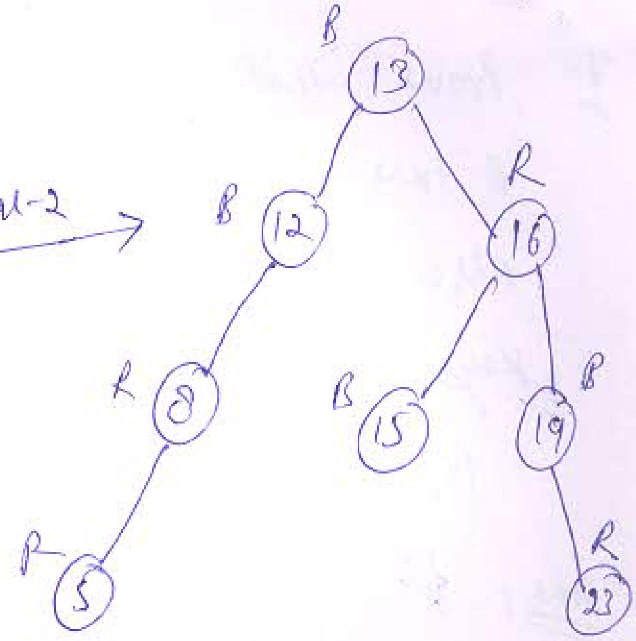
insert-5



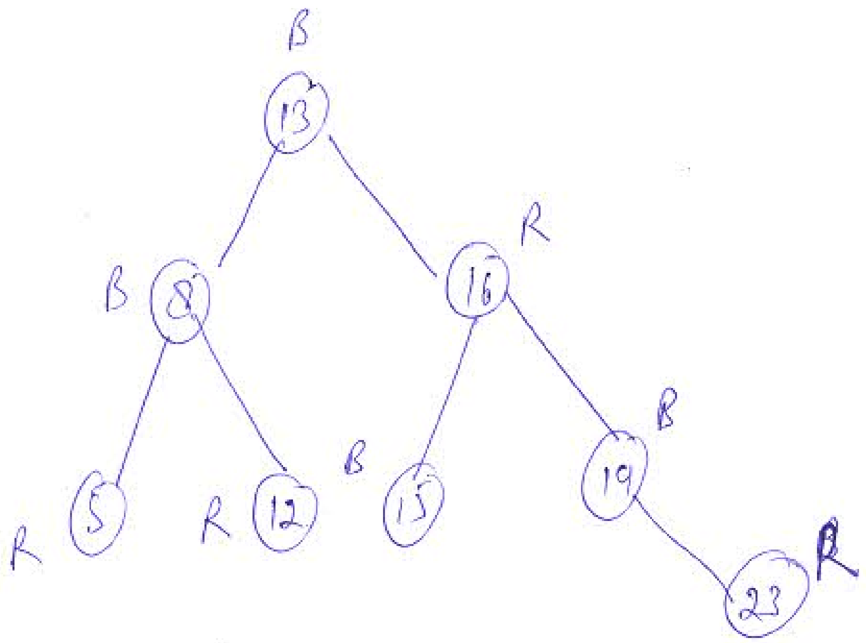
Insert-8



case-2



case-3

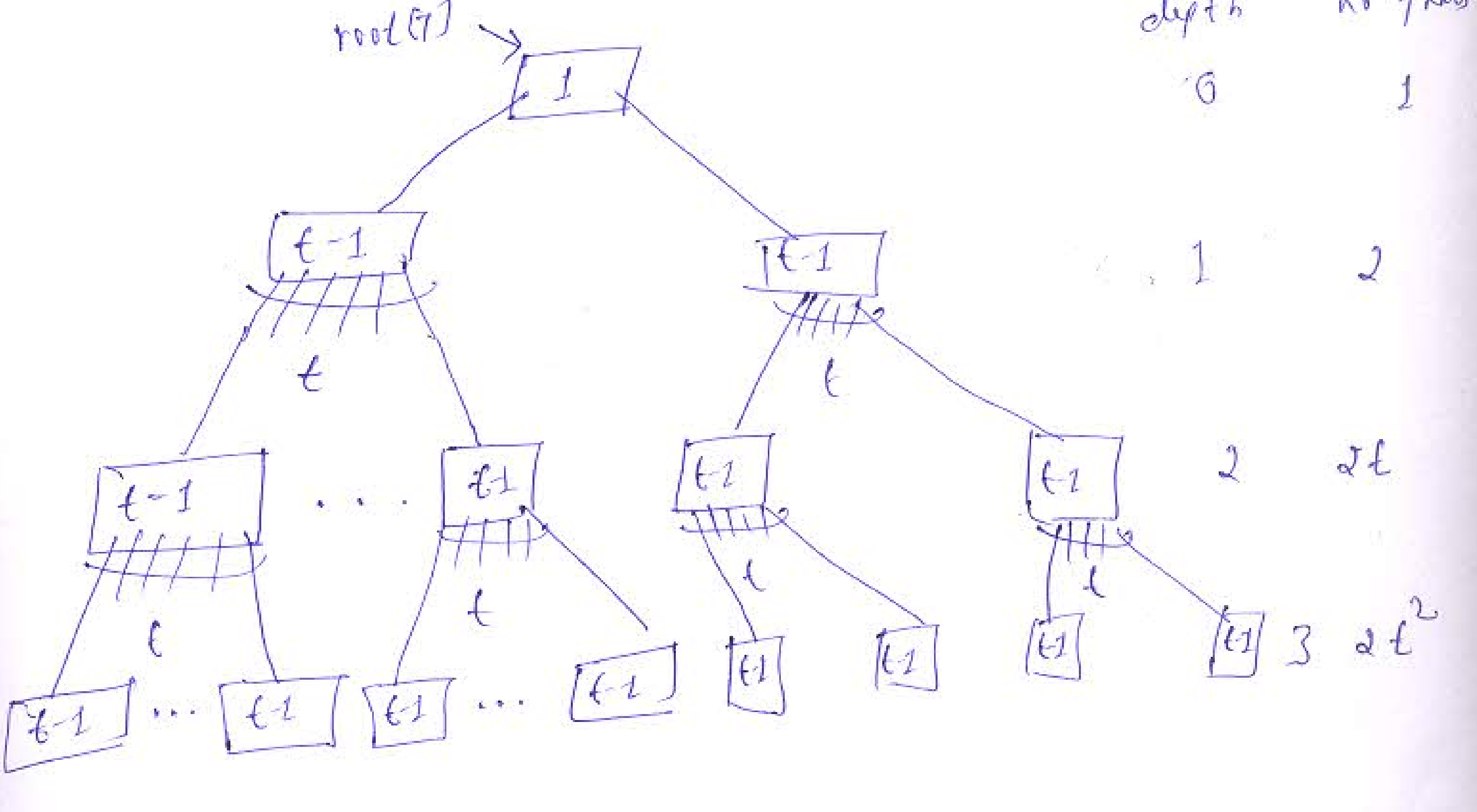


Final Red Black tree.

7. Prove that the maximum hight of any n-key B tree with minimum degree $t \geq 2$ is $\log_t \frac{n+1}{2}$. Also create a B tree ($t=2$) for the following keys

F, S, Q, K, C, L, H, T, V, W, M, R, N

Ans 1. B tree has hight h, the root contains at least one key and all other nodes contain at least $t-1$ keys. Thus there are at least 2 nodes at depth 1. at least $2t$ nodes at depth 2, at least $2t^2$ nodes at depth 3 an so on, untill at depth h there are at least $2t^{h-1}$ nodes



$$n \geq 1 + (t-1) \sum_{i=1}^{h} 2t^{i-1}$$

$$= 1 + 2(t-1)\left(\frac{t^h - 1}{t-1}\right)$$

$$n = 2t^h - 1$$

$$2t^h = n+1$$

$$t^h = \frac{(n+1)}{2}$$

$$\boxed{h = \log_t (n+1/2)}$$

F, S, Q, K, C, L, H, T, V, W, M, R, N

$t = 2$ min no of ~~node~~ Key = 1

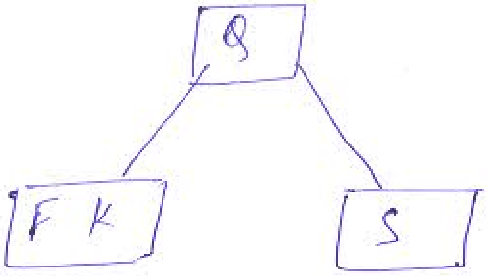
max no of Keys = 2 × 2 − 1 = 3

insert F $\boxed{F}$

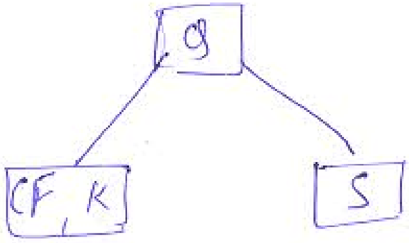
insert S $\boxed{F\ S}$

insert Q $\boxed{F\ Q\ S}$

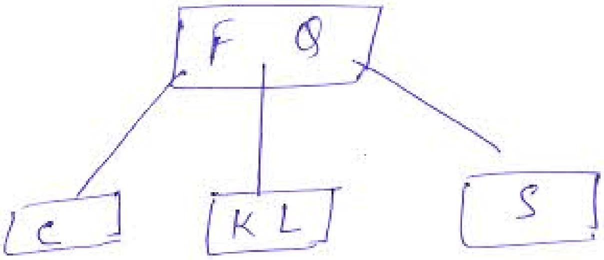
insert - K



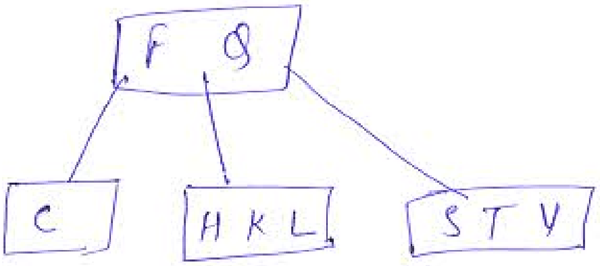
insert C



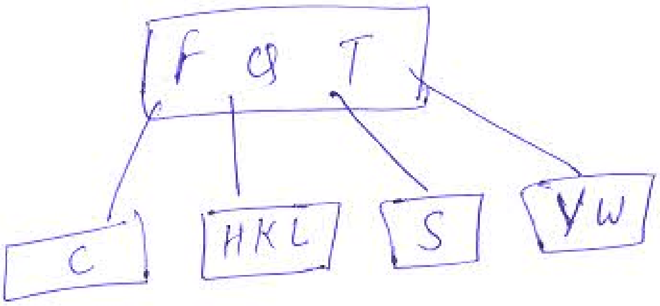
insert L



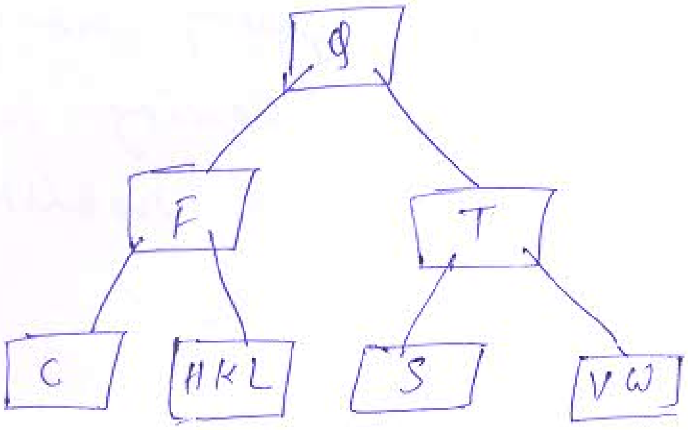
insert H, T, V

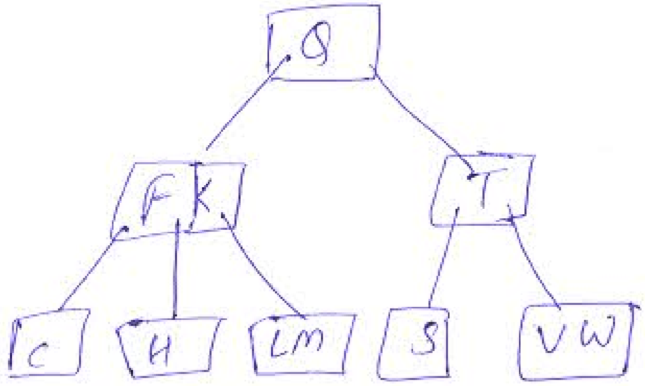


insert W

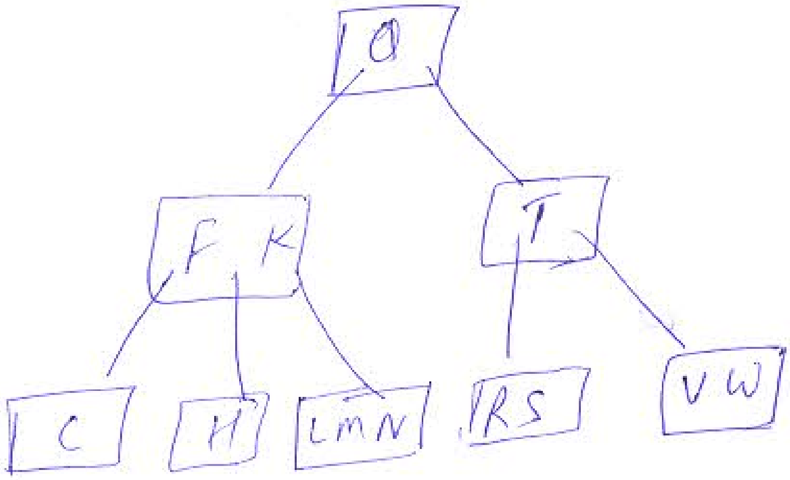


insert M





insert R, N



Q what do you mean by greedy methods? Find the solution for the following instance using fractional knapsack problem.

$$w = \{10, 20, 30, 40, 50\}$$

$$P/V = \{60, 80, 150, 120, 250\}$$

The capacity of knapsack is 80.

Ans:- A Greedy algorithm is an algorithmic paradigm that follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding a global optimum. In many problems, a greedy strategy does not in general produce an optimal solution, but nonetheless a greedy heuristic may yield locally optimal solutions that approximates a global optimal solution in a reasonable time.

vir $p = \{60, 80, 150, 120, 250\}$ $m = 80$

$w = \{10, 20, 30, 40, 50$

$P_i/w_i = \frac{60}{10}, \frac{80}{20}, \frac{150}{30}, \frac{120}{40}, \frac{250}{50}$

6, 4, 5, 3, 5

arrange the items wrt decreasing order of P_i/w_i

$P = \{60 \quad 250 \quad 150 \quad 80 \quad 120\}$

$w = \{10 \quad 50 \quad 30 \quad 20 \quad 40\}$

$x = \{1, 1, \frac{2}{3}, 0, 0\}$

$w = 10 + 50 + \frac{20}{30} = 80$

arrange in origional order

$$\boxed{x = \{1, 0, \frac{2}{3}, 0, 1\}}$$

arrange with decreasing order of p_i/w_i

$p = \{ 60 \quad 150 \quad 250 \quad 80 \quad 120 \}$

$w = \{ 10 \quad 30 \quad 50 \quad 20 \quad 40 \}$

$$W = 10 + 30 + \frac{40}{50} + 0 + 0$$

$$x = \{ 1, 1, \frac{4}{5}, 0, 0 \}$$

arrange in origional order

$$\boxed{x = \{ 1, 0, 1, 0, \frac{4}{5} \}}$$

Both Solutions are

$$\boxed{\begin{array}{l} x = \{ 1, 0, \frac{2}{3}, 0, 1 \} \\ x = \{ 1, 0, 1, 0, \frac{4}{5} \} \end{array}}$$

9. How activity selection problem is solved by greedy algorithm? Write greedy algorithm for activity selection problem.

Ans:- Let us consider Activity Selection problem using greedy method.

You are given n activities with their start and finish times. Select the maximum number of activities that can be performed by a single person, assuming that a person can work on a single activity at a time.

The greedy choice is to always pick the next activity whose time is least among the remaining activities and the start time is more than or equal to finish time of previously selected activity. We can sort the activities according to their finishing time so that we always consider the next activity as minimum finishing time activity.

Greedy - Activity - Selector (s, f)

1. $n \leftarrow$ length $[s]$

2. $A \leftarrow \{a_1\}$

3. $i \leftarrow 1$

4. for $m \leftarrow 2$ to n

5. do if $s_m \geq f_i$

6. then $A \leftarrow A \cup \{a_m\}$

7. $i \leftarrow m$

8. return A

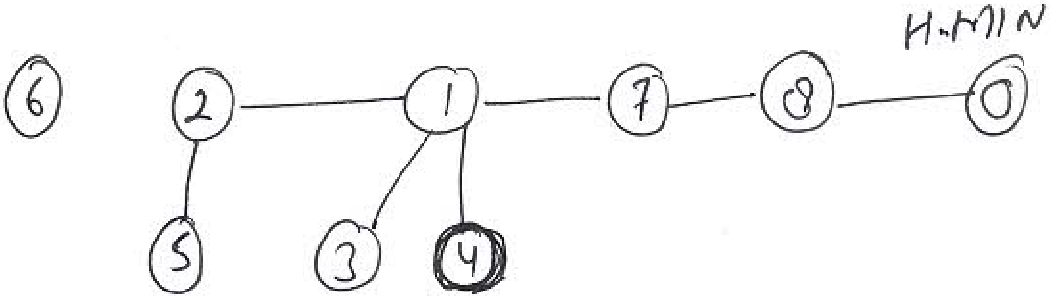
10. What do you mean by MST (minimum spanning tree)? Write an algorithm for MST that may generate multiple forest trees.

Ans 1. A minimum spanning tree is a subset of the edges of a connected, edge weighted undirected graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight. That is, it is a spanning tree whose sum of edge weights is as small as possible.

```
MST-Kruskal (G, w)
A = ϕ
for each vertex v ∈ G.V
        MAKE-SET (v)
sort the edges of G.E into indecreasing
order by weight w
for each edge (u,v) ∈ G.E taken in
    nondecreasing order by weight.
    if FIND-SET (u) ≠ FIND-SET (v)
        A = A ∪ {(u,v)}
        UNION (u,v)
return A
```

Section - C

11. Write the algorithm for consolidate operation in fibonacci Heap? also extract the minimum key of the given fibonacci heap.



Ans 1. CONSOLIDATE (H)

```
for i ← to D(n[H])
        do A[i] ← NIL
for each node w in the root list of H
        do x ← w
           d ← degree[x]
           while A[d] ≠ NIL
               do y ← A[d]
                   if key[x] > key[y]
                       then exchange x ↔ y
                   FIB-HEAP-LINK (H, y, x)
                   A[d] ← NIL
                   d ← d+1
```

$A[d] \leftarrow x$

$min(H) \leftarrow NIL$

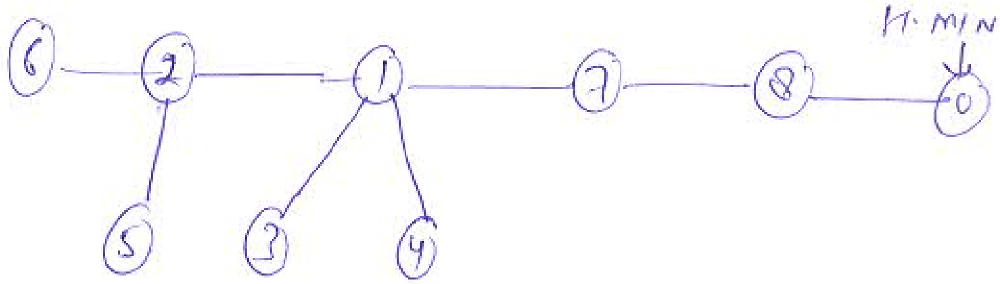
for $i \leftarrow 0$ to $D(n(H))$

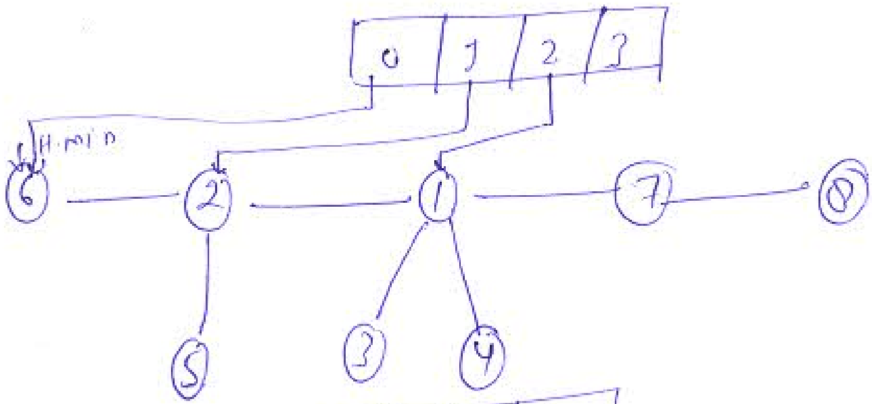
do if $A[i] \neq NIL$

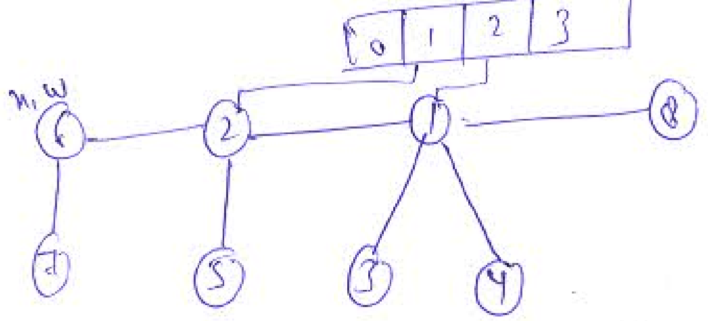
then add $A[i]$ to the root list of H

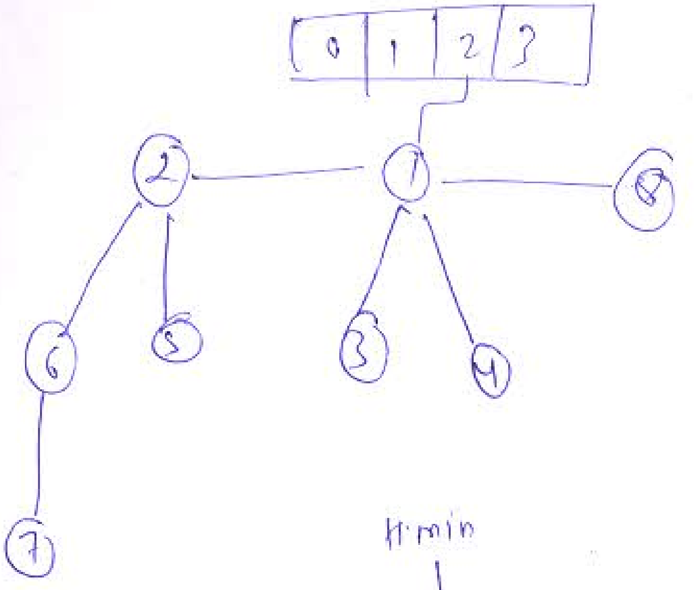
if $min[H] = NIL$ or $key[A[i]] < key(min[H])$

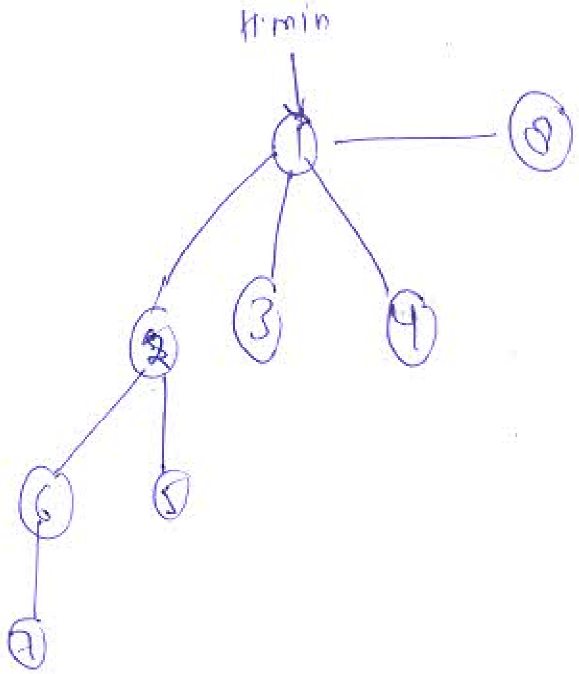
then $min(H) \leftarrow A[i]$



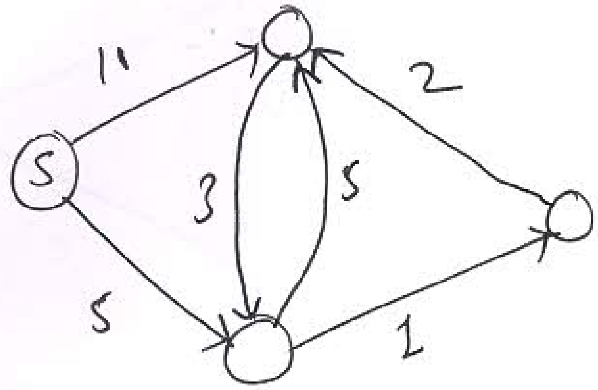








11. Given a weighted directed graph $G = (V, E)$ with source s and weight function $w: E \to R$, then write an algorithm to solve a single source shortest path problem whose complexity is $O(VE)$. Apply the same on the following graph with source S.



Ans :-

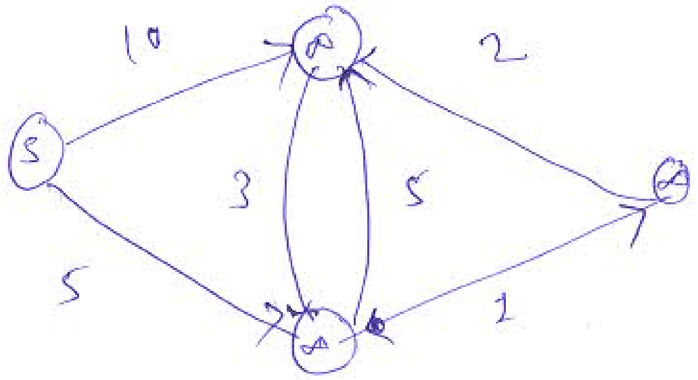
```
BELLMAN-FORD (G, w, s)
INITIALIZE-SINGLE-SOURCE (G, s)
for i = 1 to |G.V| - 1
    for each edge (u, v) ∈ G.E
        RELAX (u, v, w)
for each edge (u, v) ∈ G.E
    if v.d > u.d + w(u, v)
        return FALSE
return TRUE
```

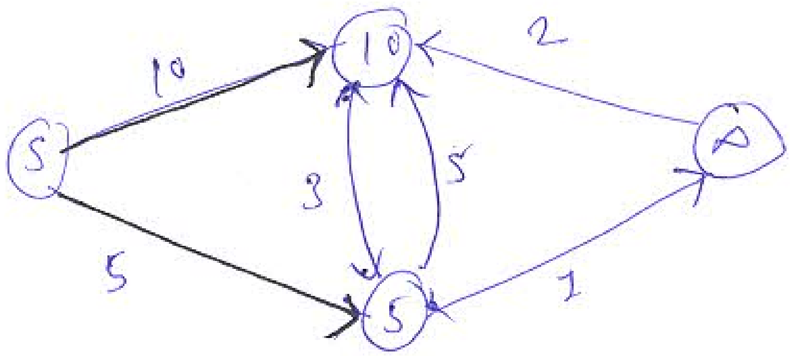
INITIALIZE - SINGLE - SOURCE (G, S)

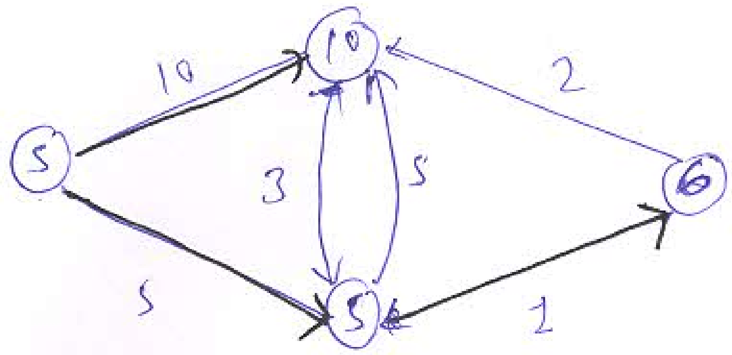
1. for each vertex $v \in G.V$
2. $v.d = \infty$
3. $v.\pi = NIL$
4. $s.d = 0$

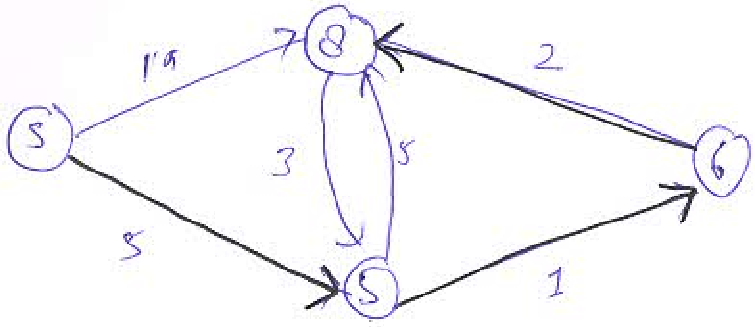
RELAX (u, v, w)

1. if $v.d > u.d + w(u,v)$
2. $v.d = u.d + w(u,v)$
3. $v.\pi = u$









Final Solution