# Ajay Kumar Garg Engineering College, Ghaziabad

## Department of MCA

### Model Solution ST-2

| | | | |
|---|---|---|---|
| Course: | MCA | Semester: | III |
| Session: | 2017-18 | Section: | MCA-1 & 2 |
| Subject: | Design & Analysis of Algorithm | Sub Code: | RCA-303 |
| Max Marks: | 50 | Time: | 2 hour |

*Note:* Answer **all** the sections.

**Section-A**

Q1. What is Red-Black Tree?

Ans. A Red-Black tree is a binary search tree with extra bit of storage information per node; its color, which can be either red or black. Red-black tree ensure that no such path is more than twice as long as any other, so the tree is approximately balanced.

Q2. What do you understand by data structure augmentation?

Ans. Augmenting a data structure is simply the process of taking an existing data structure and customizing it a little bit to fit your needs. following methodology is used:

1. Choose underlying data structure
2. Determine additional information to be stored in DS.
3. Verify that this information can be maintained for modifying operations.
4. Develop new dynamic-set operations that use the information

Q3. Write two differences between back tracking and Branch - Bound

Ans.

| Backtracking | Branch and Bound |
| --- | --- |
| ① It is used to find all possible solutions available to the Problem | ① It is used to solve optimization problem. |
| ② It traverse tree by DFS and it searches the state space tree until it found a solution. | ② It may traverse the tree in any manner, DFS or BFS. It completly searches state space tree to get optimal sol. |

Q4. What is N Queen Problem?

Ans. The N Queen is the problem of placing N chess queens on an NXN chessboard so that no two queens attack eachother. means no two ~~two~~ queens can sit in same row, same column or diagonally. N Queen Puzzle is solved by using backtracking approach.

Q5. How Greedy Method Works to Solve any Problem?

Ans. A Greedy algorithm always makes the choice that seems to be the best at the moment. This means that it makes a locally - optimal choice in the hope that this choice will lead to globally - optimal solution. In the method considering the following processes:

- Select
- feasible
- union

## Sec-B

Q6. Prove that the height of RB Tree is $2\log_2(n+1)$

Ans. first Prove by induction on height the following claim:
The subtree rooted at any node x contains at least $2^{bh(x)}-1$ nodes.
To start induction, observe that if $h(x)=0$ Then x is a leaf, so $bh(x)=1$. and indeed The subtree rooted at x contains
$2^{bh(x)}-1 = 2^1-1 = 1$ node.

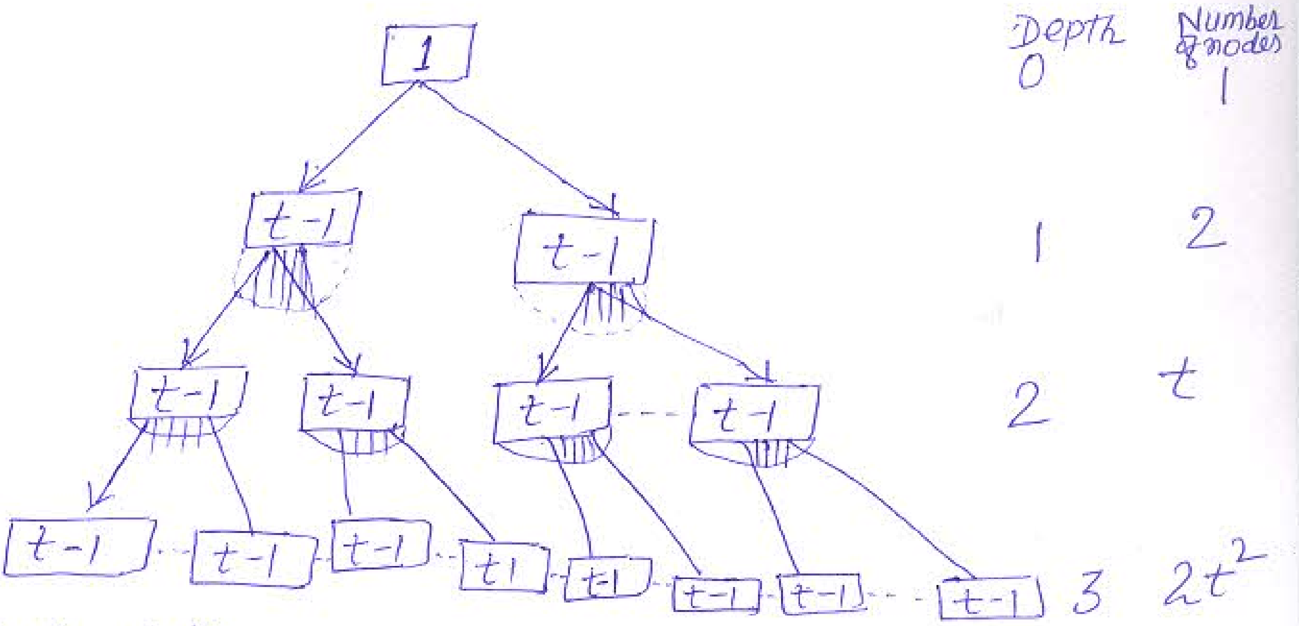
For the inductive step, consider a node x with $ht(x)>0$. Each child x has black height either $bh(x)$ or $bh(x)-1$. Applying The inductive hypothesis The subtree rooted at each child of x has atleast $2^{bh(x)-1}-1$ nodes.
Therefore, The nodes, Proving the claim
Next, let h be the height of the tree. According to property 4 at least half the nodes on the path from the root to the leaf, not including The root, must be black.
Therefore, $bh(root) \geq h/2$. Using The claim proved above, the number of nodes the tree itself is

$$n \geq 2^{bh(root)}-1 \geq 2^{h/2}-1$$

$$\Rightarrow h \leq 2\log(n+1) \Rightarrow h = O(\log n)$$

Q7. Prove that the height of B-tree is $h \leq \log_t((n+1)/2)$, if $n \geq 1$ then for any $n$-key B tree $T$ of height $h$ and minimum degree $t \leq 2$.

Ans. The root contains atleast one key. All other nodes contains atleast $t-1$ keys. There are atleast 2 nodes at depth 1, at least $2t$ nodes at depth 2, at least $2t^{i-1}$ nodes at depth $i$ and $2t^{h-1}$ nodes at depth $h$.




So we write

$$n \geq 1+(t-1)(2+2t+2t^2+2t^3+\cdots+2t^{h-1})$$

$$= 1+2(t-1)(1+t^2+t^3\cdots+t^{h-1})$$

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

$$t^h \leq (n+1)/2$$

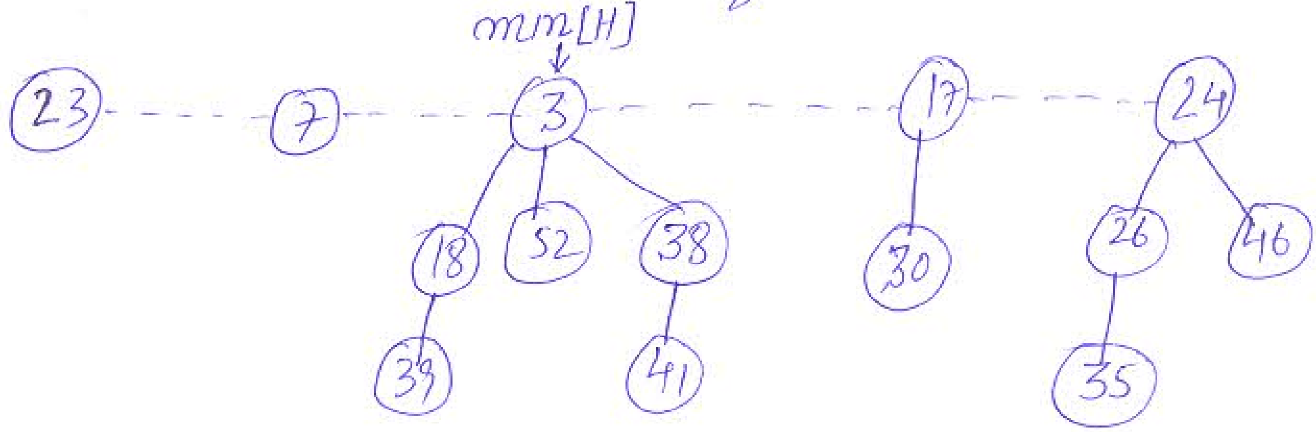
Taking base-$t$ logarithms of both sides, we get

$$h \leq \log_t (n+1)/2.$$

Q 8. What is Fibonacci Heap? Discuss its properties.

Ans. A Fibonacci Heap H is a collection of heap-ordered trees that have the following properties.

* A fibonacci heap is a Collection of min-heap ordered trees
* The trees in fibonacci heap are not constrained to be binomial trees.
* Trees within Fibonacci heap are rooted but unordered. root degrees are not unique.
* Circular doubly linked list are used in fibonacci heap
* There is a Pointer min[H] to root with minimum key that can be used to access the heap.
* A special attribute n[H] maintain the total number of nodes.
* Each node has an additional boolean label mark, indicating whether has lost a child since the last time it was made a child of another node.



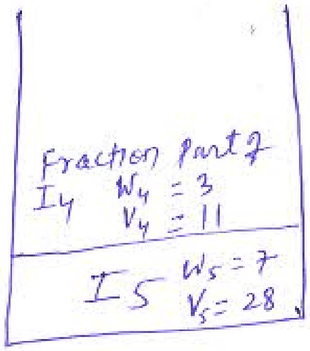
Q 9. What is 0/1 Knapsack problem? Solve the following instance using fractional Greedy approach Knapsack capacity = 10, $P = \langle 1, 6, 18, 22, 28 \rangle$ and $W = \langle 1, 2, 5, 6, 7 \rangle$.

Ans. In Knapsack Problem Weight and Values of n items are given, put these items in the knapsack of capacity W to get the maximum total value in the Knapsack. you can not break the item, either pick the complete item, or don't pick it (0-1 Property)

| Item | Weight(W) | Value(V) | $V_i/W_i$ |
|---|---|---|---|
| $I_1$ | 1 | 1 | 1 |
| $I_2$ | 2 | 6 | 3 |
| $I_3$ | 5 | 18 | 3.6 |
| $I_4$ | 6 | 22 | 3.66 |
| $I_5$ | 7 | 28 | 4 |

Now arrange the items as per their Per pound Value. (Decreasing order)

| | | | |
|---|---|---|---|
| $I_5$ | 7 | 28 | 4 |
| $I_4$ | 6 | 22 | 3.66 |
| $I_3$ | 5 | 18 | 3.6 |
| $I_2$ | 2 | 6 | 3 |
| $I_1$ | 1 | 1 | 1 |




W = 10

Thus Total Value of Knapsack

$$= 28 + 11 = 39$$

Q10. What is single source shortest paths problem? Give an algorithm for solve this problem.

Ans. Given a connected graph (weighted) $G(V,E)$, associated with each edge $(u,v) \in E$, their is a weight $w(u,v)$. The single source shortest path (SSSP) Problem is to find a shortest path from a given source $r$ to every other vertex $v \in V - \{r\}$.

Dijkstra's Algorithm:

```
Dijkstra (G, W, S)
for each vertex u ∈ V[G]
      do d[u] ← ∞
      Parent[u] ← NIL
  d[S] ← 0
  S ← φ
  Q ← V[G]
  while Q ≠ φ
     do u ← extract_Min (Q)
     S ← S ∪ {u}
     for each v ∈ adjacent[u] do
        Relax (u, v, G)
        Q ← modifykey (v)
```

---

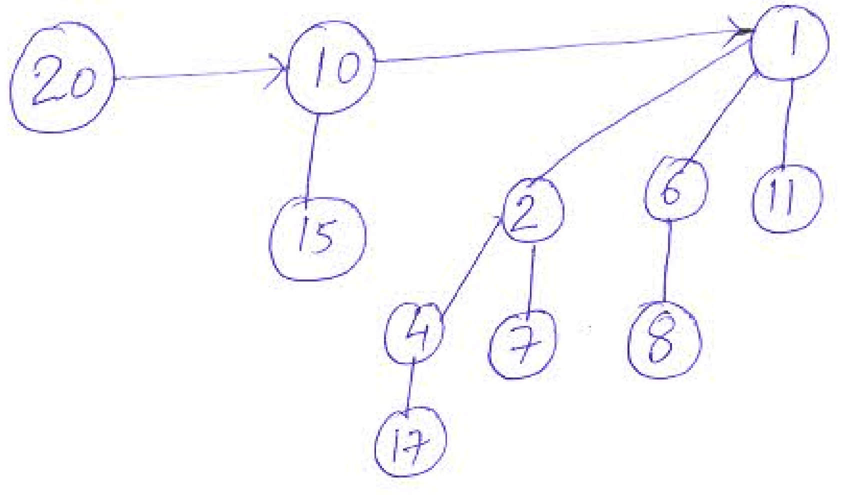
```
Relax (u, v, G)
if d[v] > d[u] + W(u, v) then
  d[v] ← d[u] + W(u, v))
  Parent (v) ← u
```

Complexity: $O(E \log V)$ when Queue is binary heap

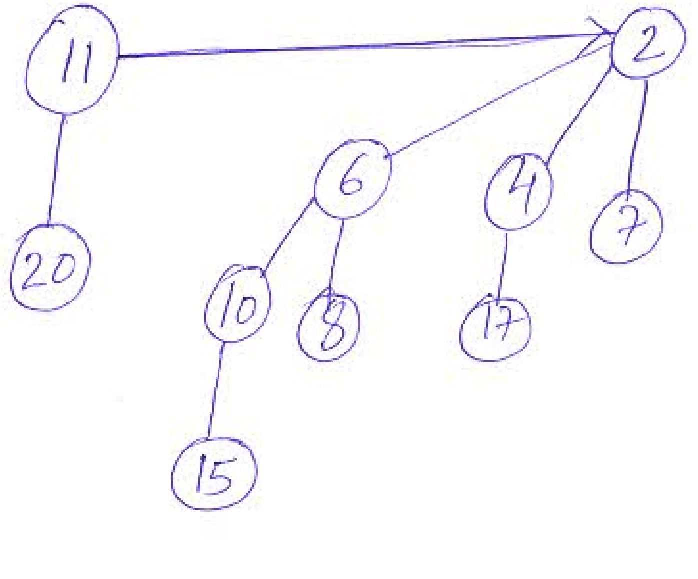
Sec - C

Q11. Construct the binomial heap for the following sequence of numbers 7, 2, 4, 17, 1, 11, 6, 8, 15, 10, 20. Also apply the operation of extracting the minimum key in the resulting binomial heap.

Ans. There are total 11 elements binary of 11 → $\overset{b_3}{1}0\overset{b_1}{1}\overset{b_0}{1}$

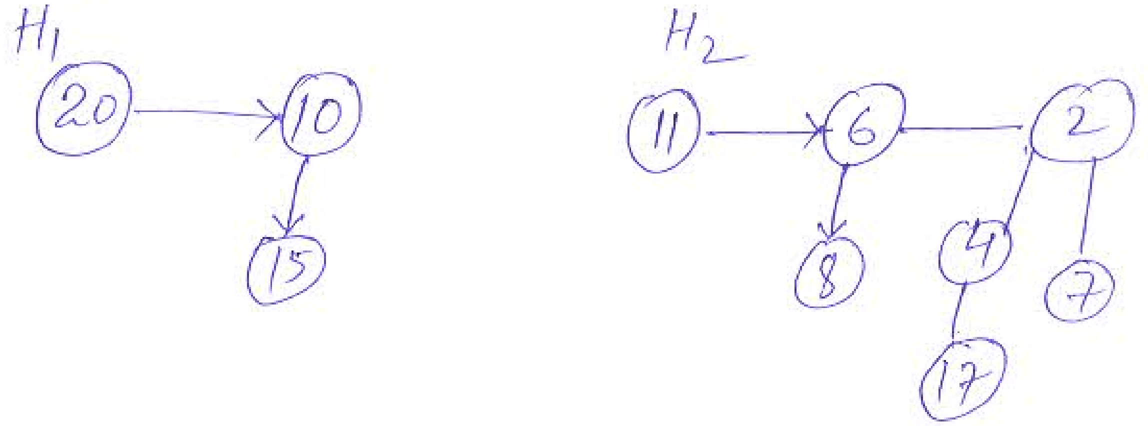


Extracting min key

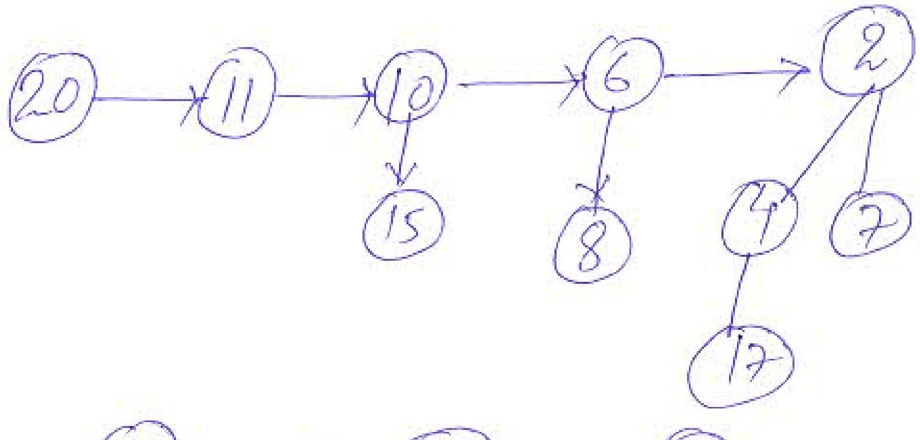


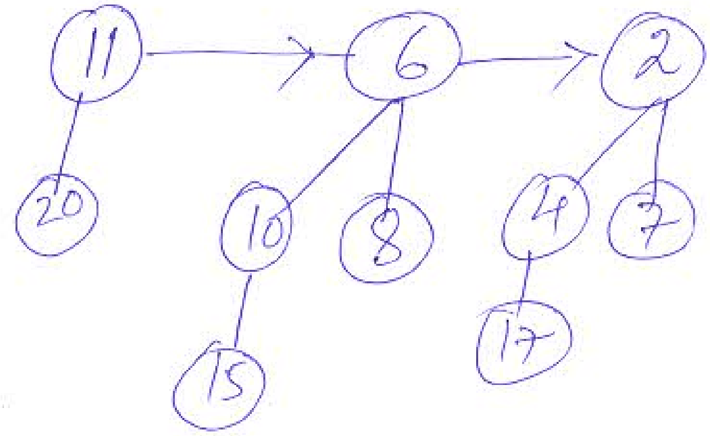
Steps in extracting min:

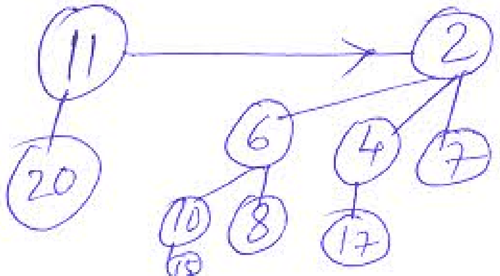
Step 1. Find and extract the minimum element from the top root nodes and reverse all the dangling forests.




Step 2. Union($H_1$, $H_2$)



Q12. What do you mean by Minimum spanning Tree? Write an algorithm that always generate single forest tree

Ans. A minimum spanning tree is a spanning tree with weight less than or equal to weight of every other spanning tree.

Kruskal's Algorithm:

```
Kruskal()
{
  T = ϕ
  for each v ∈ V
   Makeset(V);
  Sort E by increasing edge weight w
  for each (u, v) ∈ E (in sorted order)
   if Findset(u) ≠ Findset(V)
     T = T ∪ {{u, v}},
    Union (Findset(u), Findset(V));
   }
```

Complexity $O(E \lg E)$