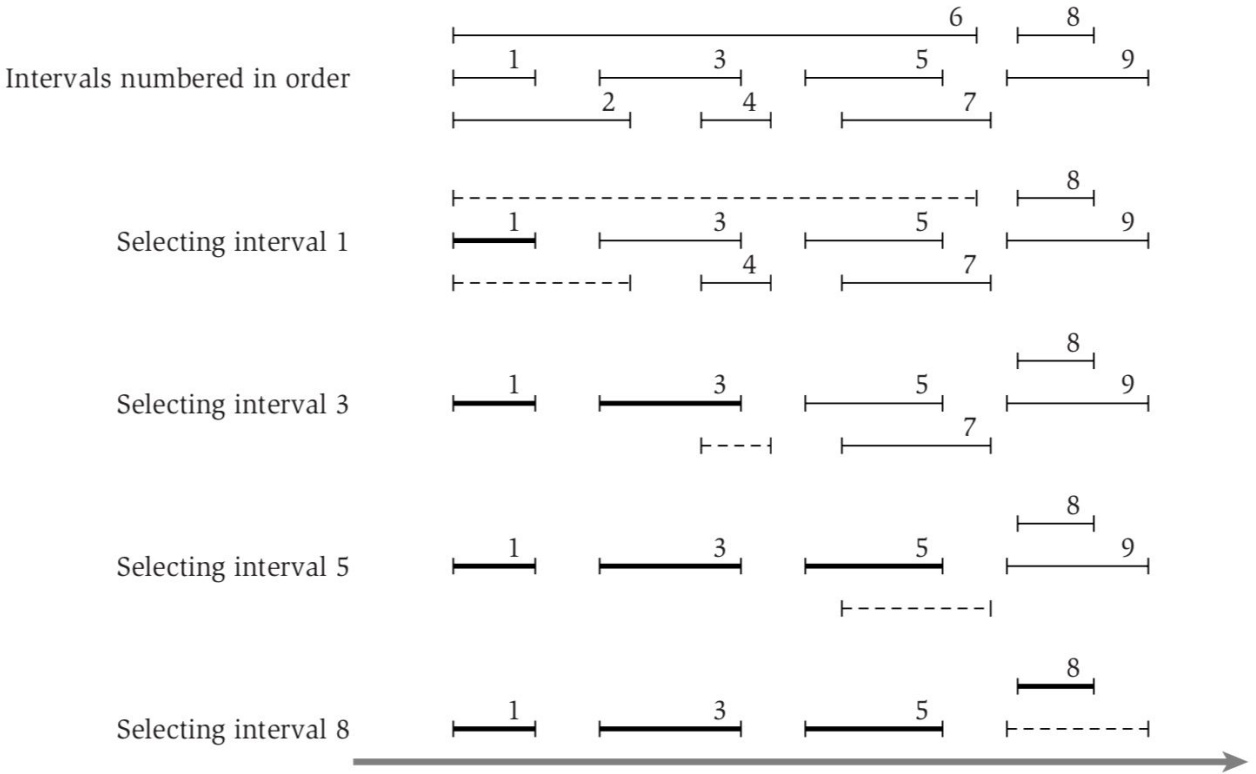
CS180 Discussion

Week 4

Lecture Recap

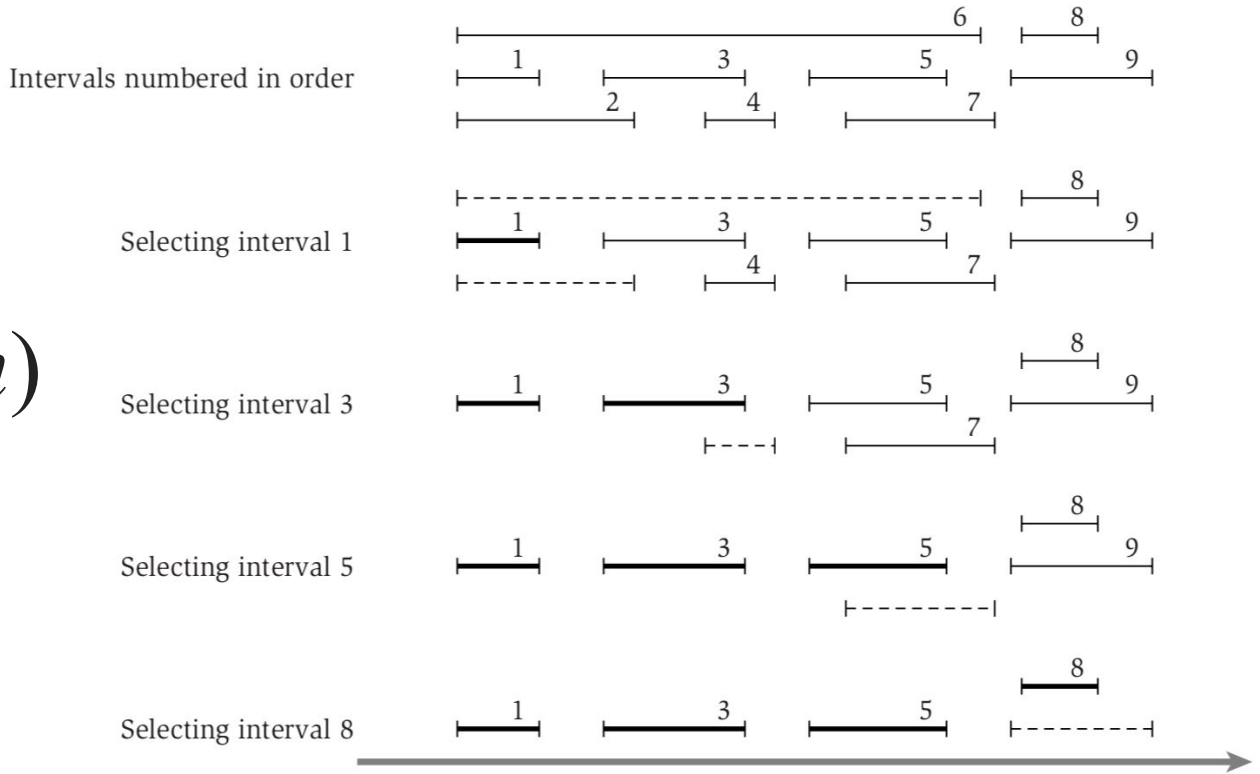
- Greedy algorithms
- Interval scheduling
- Heaps
- Dijkstra shortest path
- Minimum spanning trees
- Prim's algorithm
- Kruskal - not explained

Interval scheduling



Interval scheduling

$O(nlogn)$

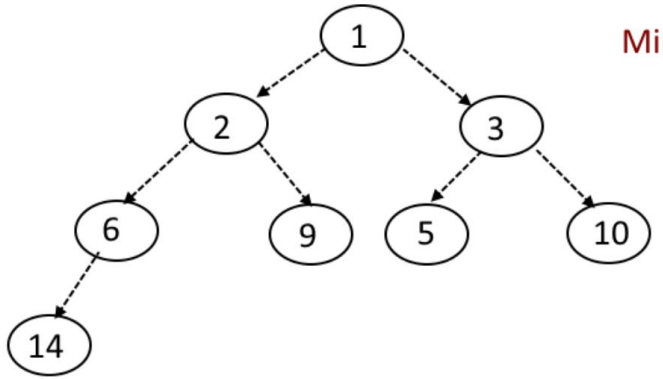


Heaps

A Binary Heap is a Binary Tree with following properties.

1) It's a complete tree (All levels are completely filled except possibly the last level)

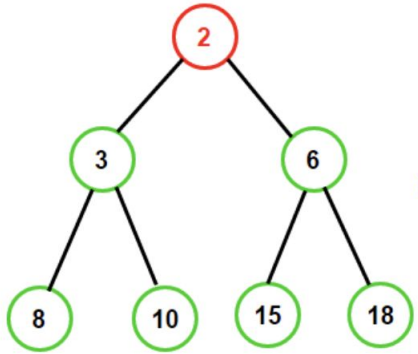
2) A Binary Heap is either MinHeap or MaxHeap. In a Min Binary Heap, the key at root must be minimum among all keys present in Binary Heap. The same property must be recursively true for all nodes in Binary Tree. MaxHeap is opposite to MinHeap.



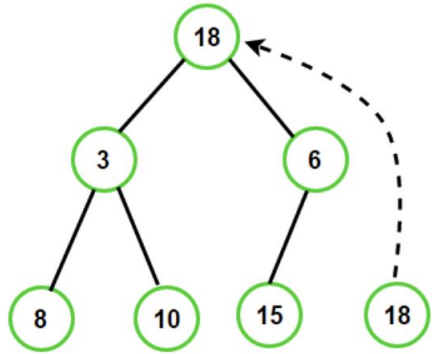
0	1	2	3	4	5	6	7	8
	1	2	3	6	9	5	10	14

for Node at i : Left child will be 2i and right child will be at 2i+1 and parent node will be at [i/2].

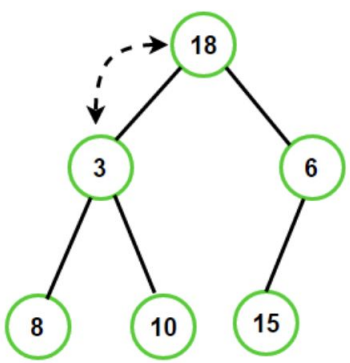
Heaps



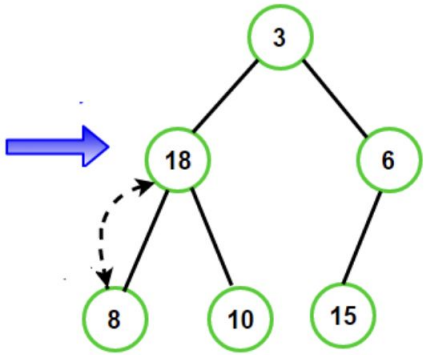
Pop() called on min heap



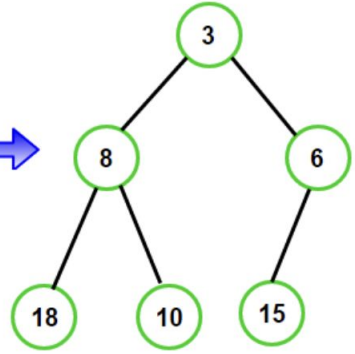
Replace the root of the heap with the last element on the last level and call
Heapify-down(root)



Swap root node with its smaller child
swap(18, min(3, 6))

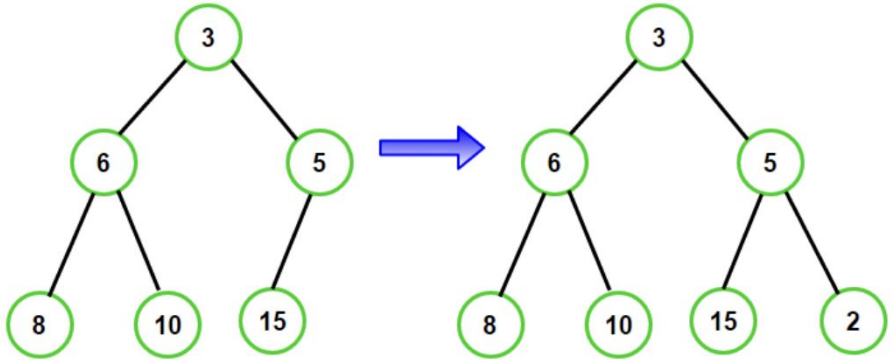


Swap node 18 with its smaller child
swap(18, min(8, 10))



Resultant Min Heap

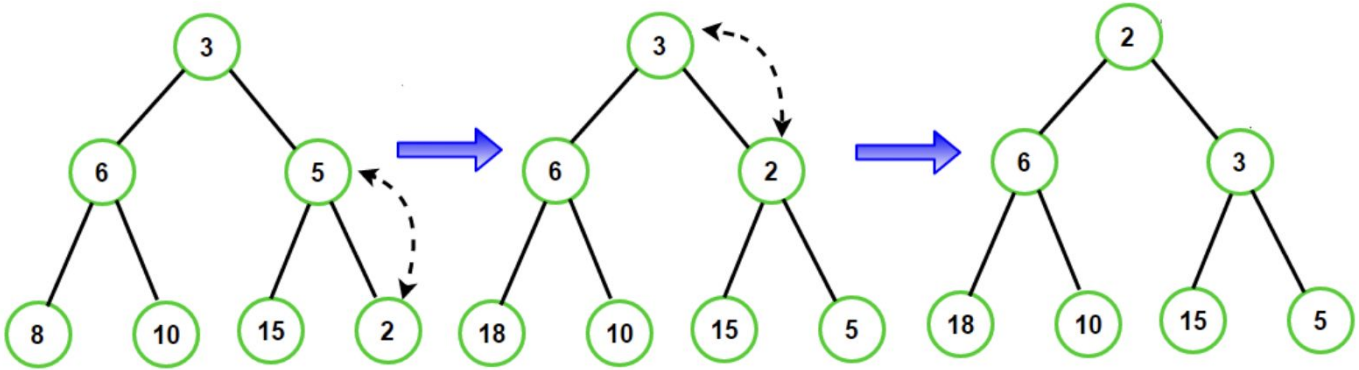
Heaps



Push(2) called on min heap

Add the new element 2 to the bottom level of the heap and call

Heapify-up(2)



Swap node 2 with its parent as heap property is violated

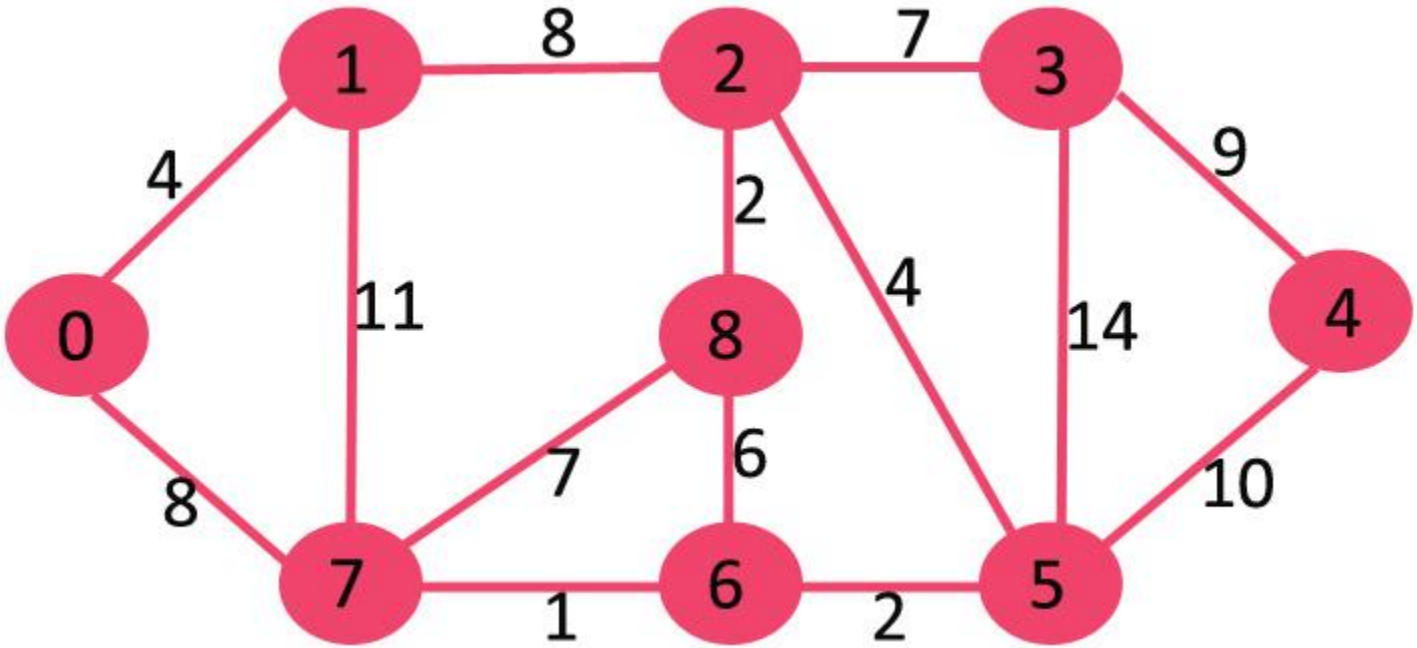
swap(5, 2)

Swap node 2 with its parent as heap property is still violated

swap(3, 2)

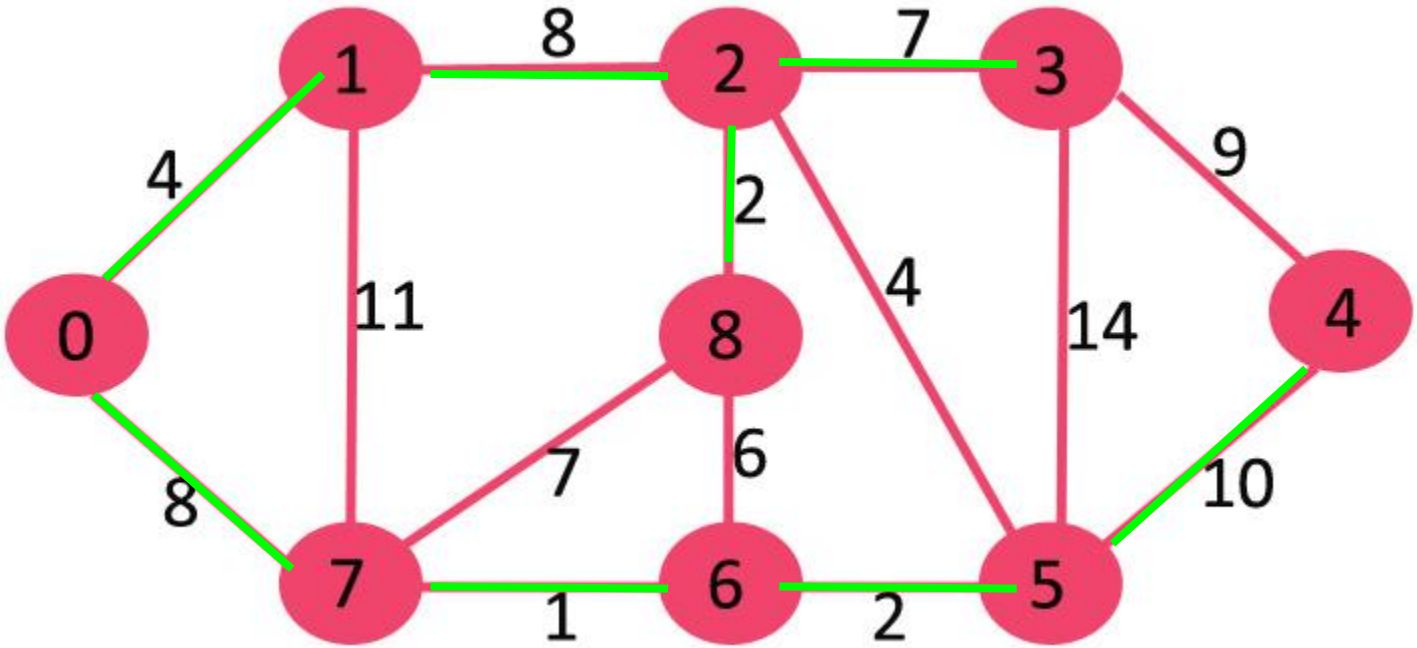
Resultant Min Heap

Graph Spanning Algorithms



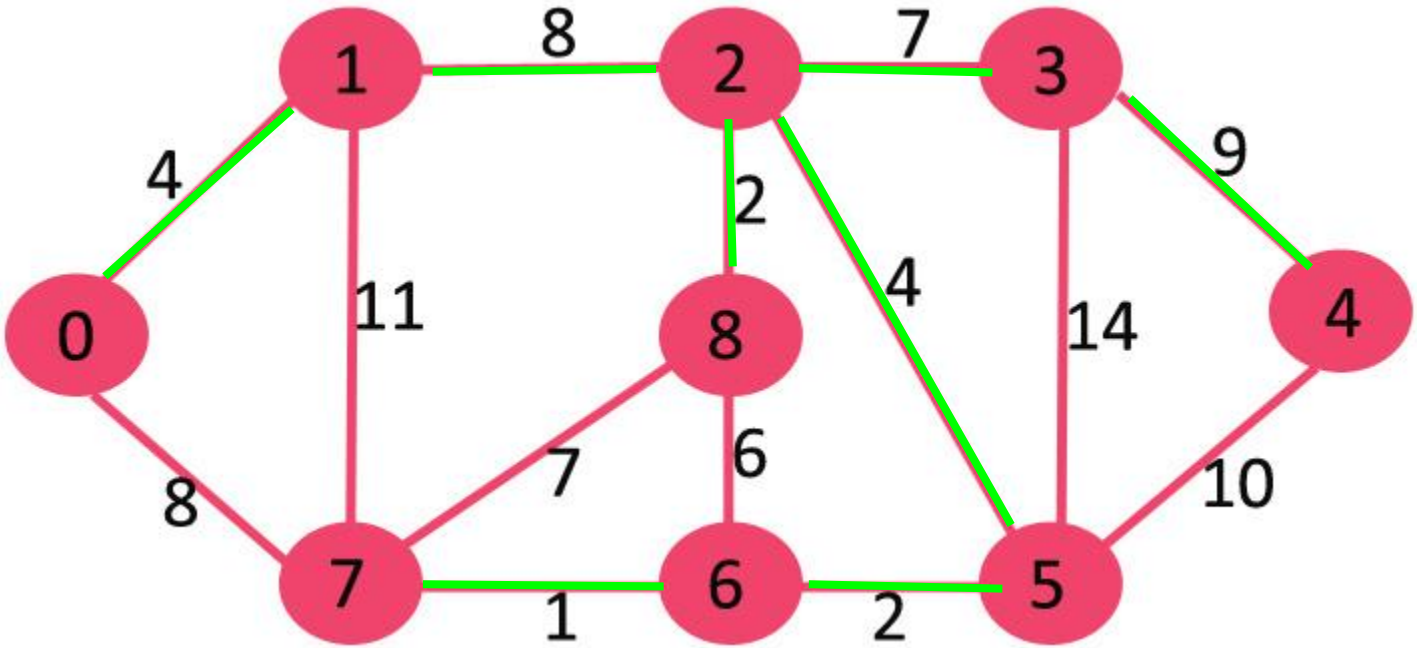
[Prim]
[Dijkstra]

Graph Spanning Algorithms



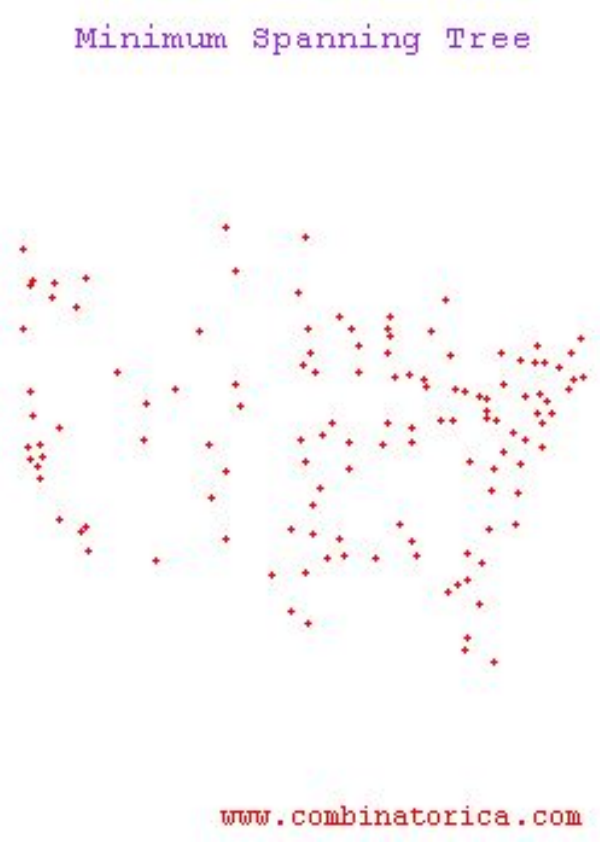
[Prim]
[Dijkstra]

Graph Spanning Algorithms



[Prim]
[Dijkstra]

Graph Spanning Algorithms



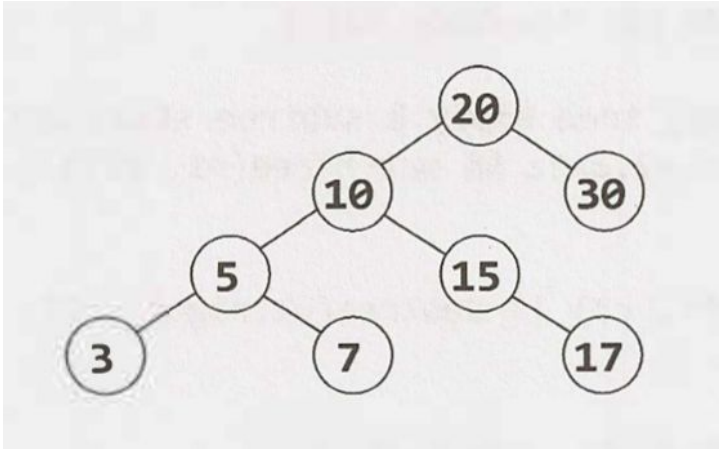
Kruskal

Interview Questions!!!

Interview Questions!!!
Interview Questions!!!
terview Questions!!!

getRandomNode

Random Node: You are implementing a binary search tree class from scratch, which, in addition to insert, find, and delete, has a method getRandomNode() which returns a random node from the tree. All nodes should be equally likely to be chosen. Design and implement an algorithm for getRandomNode(), and explain how you would implement the rest of the methods.



Paths with Sum

You are given a binary tree in which each node contains an integer value (which might be positive or negative). Design an algorithm to count the number of paths that sum to a given value. The path does not need to start or end at the root or a leaf, but it must go downwards (traveling only from parent nodes to child nodes).