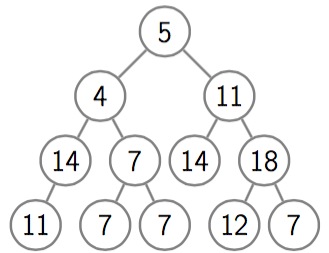
1. How many edges of this binary tree violate the min-heap property?



1. Which of the following arrays is a heap

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | 10 | 15 | 8 | 20 | 14 | 17 | 12 | 9 | 13 |
| b | 100 | 65 | 8 | 40 | 34 | 7 | 2 | 9 | 13 |
| c | 100 | 65 | 45 | 30 | 28 | 23 | 7 | 5 | 1 |

1. Write a C++ function to check whether a given array is max heap or not.
2. Given a heap write down two functions to return the maximum and minimum values, and analyze their complexity.
3. Show the heap after removing the first five elements from the following array (after removing each maximum the array is heapified)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 100 | 65 | 8 | 40 | 34 | 7 | 2 | 9 | 13 |

1. Consider a complete binary tree represented by an array [19,14,28,15,16,7,27,15,21,21,5,2], How many edges of this tree violate the max-heap property?
2. Assume that a max-heap with 10^5 elements is stored in a complete 5-ary tree. Approximately how many comparisons a call to Insert() will make?

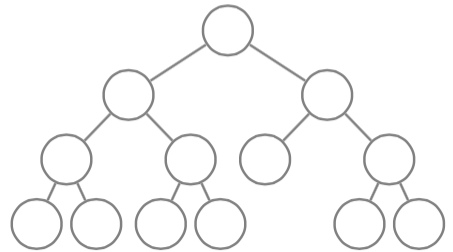
//solution: log(10^5) to the base of 5 (7.153 which is approximately equal to 8 comparisons).  
We will need to make 8 sift-up operations

1. Assume that a max-heap with 10^6 elements is stored in a complete 7-ary tree. Approximately how many comparisons a call to ExtractMax() will make?

solution : Zero, finding the maximum is an O(1) operation if considering it returns max only without poping it

It depends on what ExtractMax is doing.  
If it’s popping then the complexity is emmm 7 \* log(10^6) to the base of 7   
We will extract the maximum of the child nodes log(10^6) to the base of 7 times.  
And in each time we compare it to each of its 7 children.

1. This binary tree contains 13 nodes, and hence we have 13 subtrees here (rooted at each of 13 nodes). How many of them are complete? solution:11



**Optional Heap Problems:**

1. Jessie and Cookies - HR - [link](https://www.hackerrank.com/challenges/jesse-and-cookies/problem).
2. Find the Running Median - HR - [link](https://www.hackerrank.com/challenges/find-the-running-median/problem).
3. Minimum Average Waiting Time - HR - [link](https://www.hackerrank.com/challenges/minimum-average-waiting-time/problem).
4. Vile Grasshoppers - CF - [link](http://codeforces.com/contest/937/problem/B).