CS 3130 **Design and Analysis of Algorithms** **Fall 2018**

K E Y T O Q U I Z #8

[6 points]

---

1. [2 points] Indicate time efficiency of each of the following algorithms (as a function of the size of its input n):

(a) Top-down construction of a heap: $\boxed{O(n \lg n)}$ (b) Heapsort: $\boxed{\Theta(n \lg n)}$

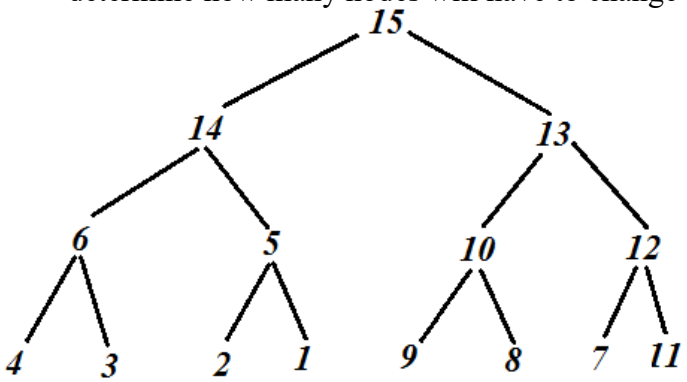
(c) Bottom-up construction of a heap: $\boxed{O(n)}$ (d) MAX-HEAPIFY function: $\boxed{O(\lg n)}$

2. [1 point] Assumi that a priority queue is implemented as a max-heap, indicate time efficiency of the following two operations:

(e) Inserting an item into a priority queue: $\boxed{O(\lg n)}$;

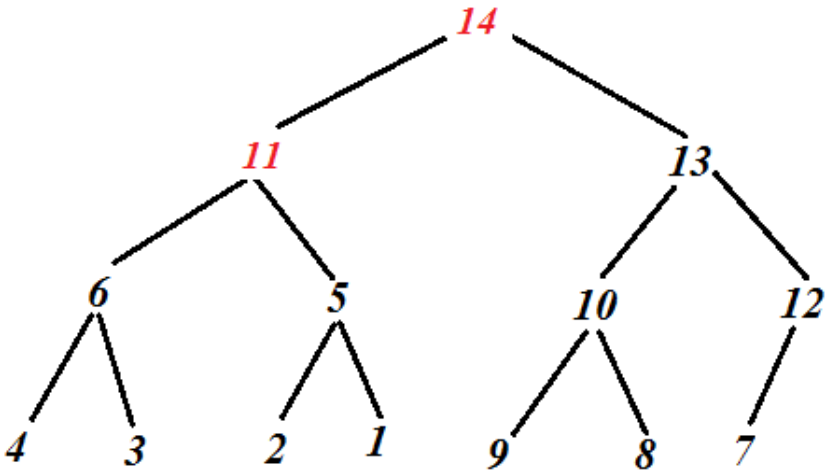
(f) Removing an item with the highest priority from a priority queue: $\boxed{O(\lg n)}$.

3. [3 points] Show graphically how to remove the item with the highest priority from the heap below and determine how many nodes will have to change their positions during this process:



Answer:

The result will be the following heap:



Only 2 nodes had to be moved when a root value is removed