CS 1332 Recitation 5 Worksheet – Heaps, Build Heap

CS 1332 TAs

February 11, 2019

This worksheet covers material from this week's recitation. It is meant to be additional exercise for your benefit and will not be graded. Feel free to collaborate with other students on this and ask TAs for help. Distribution of this worksheet is **not** permitted.

1 Complexity Questions

1. Runtime of adding to a min-heap: _____

2. Runtime of removing from a max-heap: _____

3. Runtime of adding to a min-priority queue backed by a min-heap: _____

4. Runtime of running build heap: _____

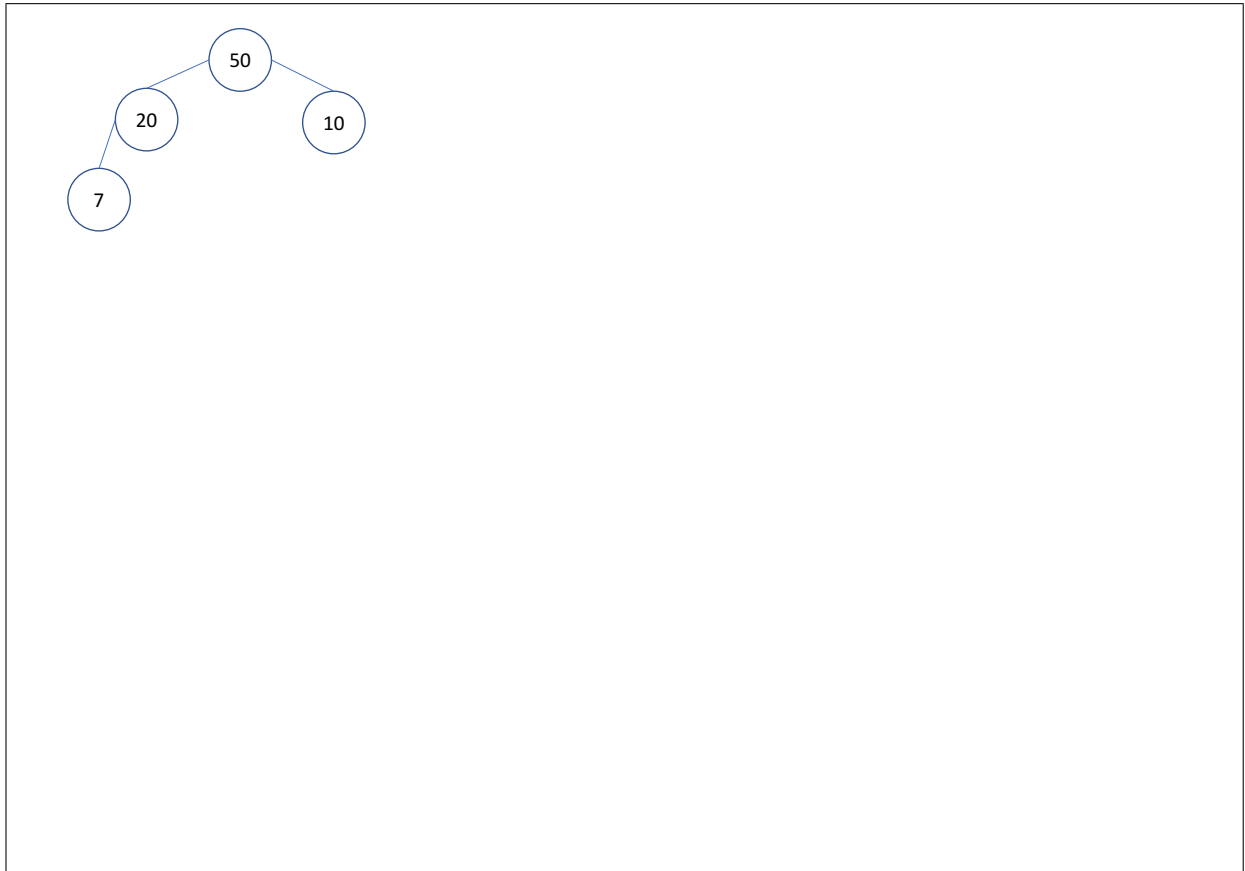5. Runtime of converting a min-heap into a max-heap: _____

2 Tree Identification

Pick the most accurate description for each tree out of: Tree, Complete Tree, Min-Heap, Max-Heap.
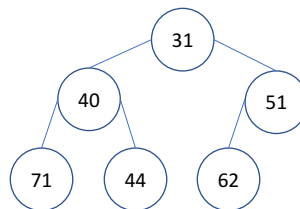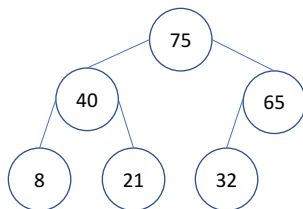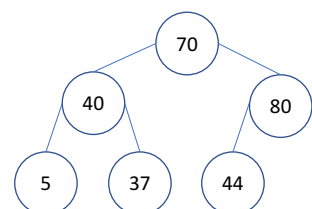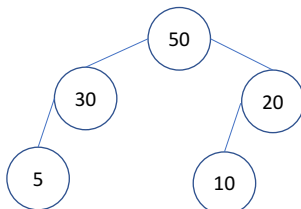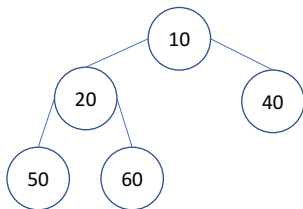
3   Diagramming

1. Perform the following operations on the max-heap provided below and draw the heap after each operation:
   `add(40), add(90), remove(), add(11), remove().`

2. Run build-heap on this array to create a min-heap

0	1	2	3	4	5	6	7	8	9	
	4	80	70	2	30	20	7	26	8	Start

4 Above and Beyond

This is a small set of challenge questions related to the material in the course, similar to those that you may find in a Software Engineering interview. Again, feel free to collaborate with other students or ask a TA for help.

1. Given an arbitrary array, write an `isHeap()` method that returns true if the array is a valid max-heap and false otherwise. Assume index 0 is unoccupied.

   Examples:
   Input: $[X, 1, 2, 3, 4, 5, 6]$ Output: true

   Expected time: $O(n)$, Expected space: $O(1)$ outside of original array

   ```
   public boolean isHeap(int[] arr) {

   }
   ```

2. Write a function `kSmallest()` that returns the k smallest elements in an arbitrary array in descending order using a heap.

   Example:
   Input: $[4, 1, 2, 5, 8, 6]$, $k = 3$
   Output: $[4, 2, 1]$
   Expected time: $O(n \log k)$, Expected space: $O(k)$ outside of original array.

   ```
   public int[] kSmallest(int[] arr, int k) {

   }
   ```