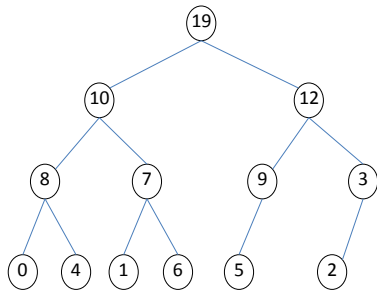


Module IN2002—Data Structures and Algorithms

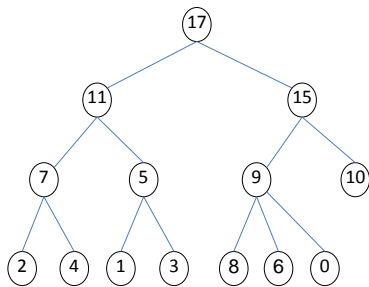
Exercise Sheet 3

1. Only one of the following trees is a heap. Indicate which one and why the others are not.

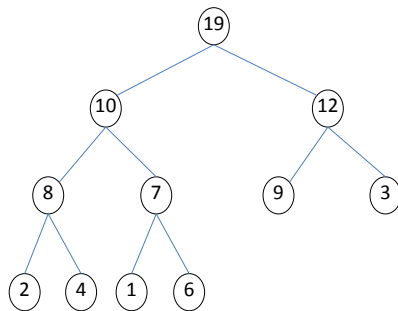
a)



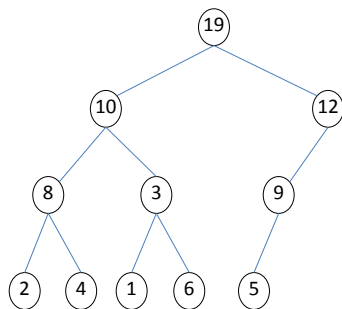
b)



c)



d)



2. Show the heaps that are generated as you add the following sequence of keys to an empty heap (one at a time): 6, 3, 11, 9, 8, 5, 4, 12.

3. Only one of the following arrays is not a heap. Indicate which one.

- a) 12 11 8 10 3 4 6 7 5 1 2
- b) 30 17 16 15 14 3 2 8 11 7 6 5
- c) 20 8 14 7 1 5 10 3 6

4. Consider heap 16 14 10 8 7 9 3

- a. Show it in tree format.
- b. Show the heaps that result if `extractMax` is applied repeatedly until the heap is empty.

5. Provide pseudocode for a queue implemented using an array. This implies the functions *isEmpty*, *enqueue*, and *dequeue*.

6. Write functions `void add(int elt)` and `int extractMax()` in Java, implementing the pseudocode in the lecture notes.

7. Work through the steps of sorting the keys 7 4 1 8 5 2 9 6 3 0 using heap sort. Focus on the major steps: show the state of the array after each insertion and extraction from the heap.

And a bit of programming. You are not expected to tackle this during the tutorial slot, but at some later time, at your own convenience. Note that answers to this will be released much later than for the other questions, giving you time to experiment with it.

8. Write a recursive program with the declaration

```
public static void permute( String str );
```

that prints all the permutations of the characters in the string `str`. If `str` is "abc", then the strings output are abc, acb, bac, bca, cab, and cba.