**Priority Queues (Course 2 Week 3 Quiz) (5/6 83% on first attempt)**

1. Min Heap property – (5 – 4), (14 – 11), (18 – 12), (18 – 7). Total : 4
2. 0 on level 1, 3 on level 2, 1 on level 3. So 4 subtrees are complete. Note that subtrees rooted at the leaves are complete.

19

14 28

15 16 7 27

15 21 21 5 2

1. Max Heap property violation: (19 – 28), (14-15), (14-16), (15-21), (16-21). Total: 5
2. Insert operation uses siftUp – which is O( log\_5 (10^5)) = 7.15338 = 8 (rounded up)
3. ExtractMax() calls siftDown –hence no. of comparisons would be

O(7 \* log\_7 (10^6)) = 49.698 = 50

1. Sry too much math I lazy

Ooff this was correct!!

**Disjoint Sets (Course 2 Week 3 Quiz) (4 / 4 - 100% on first attempt)**

1. Refer to working below.

There are 12 nodes – all singleton sets.

Union(2,10) => 2 and 10 are in a set

Union(5,7) => 5 and 7 are in a set.

Union(6,1) => 1 and 6 are in a set.

Union(3,4) => 3 and 4 are in a set.

Union(5,11) => 5, 7, 11 are in a set

Union(7,8) => 5, 7, 8, 11 are in a set

Union (7,3) => 3, 4, 5, 7, 8, 11 are in a set

Union (12,2) => 2, 10, 12 are in a set.

Union(9,6) = > 1,6,9 are in a set.

Final Sets

3,4,5,7,8,11

2,10,12

1,6,9

Find(6) = 1,

Find(3) = 3

Find(11) = 3

Find(9) = 1

Output: **1 3 3 1**

1. Refer to the working below.

There are 12 nodes.

Union(2,10) => 2 and 10 are in a set (Height:1)

Union(5,7) => 5 and 7 are in a set. (Height: 1)

Union(6,1) => 1 and 6 are in a set. (Height: 1)

Union(3,4) => 3 and 4 are in a set. (Height: 1)

Union(5,11) => 5, 7, 11 are in a set (Height: 1)

Union(7,8) => 5, 7, 8, 11 are in a set (Height: 1)

Union (7,3) => 3, 4, 5, 7, 8, 11 are in a set (Height: 2) – Both trees are of same height – so height increases by 1)

Union (12,2) => 2, 10, 12 are in a set. (Height: 1)

Union(9,6) = > 1,6,9 are in a set. (Height: 1)

In the resulting forest, there are 3 subtrees.

Product of heights = 2 x 1 x 1 = **2**

1. Refer to the working below.

Let i = 3

There will be 3 nodes

Union(1, 2) => 1 and 2 in same set (height 1)

Union(2, 3) => 1,2,3 in the same set.

Total 1 tree of height 1

Generalising in terms of n,

We still get **1 tree of height 1.**

1. Refer to working below.

Since Find() is called from nodes 1 to 60, and the root can never be compressed, the maximum height after 60 calls to Find() with path compression is **1**.