CS2210A - Assignment 3
24 October 2018
Syed Ahmed
250897473 – Class ID: 3

Question 1

0	1	2	3	4	5	6
	22			18	12	
				11	47	

Question 2

0	1	2	3	4	5	6
47	22			18	11	12

Question 3

0	1	2	3	4	5	6
47	11		22	18	12	

Question 4

$f(0) = c_1$
$f(n) = f(n–1) + c_2n + c_3$, for $n > 0$

Need to start by unwrapping, so
$f(n) = f(n–1) + c_2n + c_3$
$\quad = f(n-1-1) + c_2n + c_2n + c_3 + c_3$
$\quad = f(n-1-1-1) + c_2n + c_2n + c_2n + c_3 + c_3 + c_3$
$\quad$ **$= f(n-1*i) + i*c_2n + i*c_3$**

Unwrapping will stop when $n = 1*i$, or equivalently, **when $i = n$**

Thus, $f(n) = f(0) + n*c_2n + n*c_3$
$\qquad$ **$= c + n^2c_2 + n*c_2$**

Therefore, the time complexity in big-Oh notation is $O(n)^2$

Question 5
i)

Pseudocode for algorithm as described:

```
Input: root r of tree
Output: max integer of tree, stored in variable v

Algorithm maxValue(r)
     set integer max equal to r
     initialize queue q
     set integer v equal to zero
     add r to queue
     while the q is not empty
          set integer temp equal to q.remove
          if temp.value is greater than v
               set v equal to temp.value
          if temp.isLeaf returns true
               continue
          else
               for each child c of v do
                    add c to q
     return v
```

ii)

First, need to analyze algorithm without recursive calls

c operations in base case

$c_3 + c_2$ x degree(r) in recursive case

Next, need to find number of recursive calls per node

1 per node

Then, count total number of operations

$\sum_{\text{leaves(n)}} c_1 + \sum_{\text{internal(u)}} (c_3 + c_2 \text{ degree(u)})$

c_1 x no. of leaves + c_3 x no. of internal + $c_2 \sum_{\text{internal(u)}}$ degree(u)

$\cancel{c_1}$ x no. of leaves(n) + $\cancel{c_3}$ x no. of internal + $\cancel{c_2}$(n-$\cancel{1}$)

No. of edges = n – 1

Therefore, worst case time complexity is O(n)