CS323 Homework Assignment 1

Yibo Wang

January 18, 2017

Collaboration Statement: This is my own work. I worked on it by myself without consulting anyone else.

1. Explanations for ComplexCode4:

First, let's note that this code has two nested for-loops. The first for loop goes from int i = 1 to when int 1 is less than N and increases by a factor of 2.

The second for loop goes from int j = 0 until when j is less than some number N and the overall loop increments by a factor of 2 which can be shown by the math equation 2^n. Therefore the overall asymptotic complexity of the code is log n times N.

2. Explanations for ComplexCode5:

The first for-loop decreases by 1 so it goes from some number N to 0. The second for-loop goes from 100 to some number N. The last for-loop goes 15 to some number N*2 but increases by 5. The overall asymptotic complexity is therefore N^3 based on the three for loops.

Overall this code is because the two while loops cause the iterator to move in a second order growth instead of in a linear growth.

3. Explanations for ComplexCode17:

The solution is N^2. Since there is a for-loop the first loop goes from int i when it is zero to when i is less than some number N so this means that asymptotic complexity is at least N. The next item to consider is that the if-clause which says !x.contains the array index. This cause is going to add an integer to the arraylist from integers 0 to N since the command checks if the ArrayList x contains the element in the arraylist. So the two factors to consider are the first for loop that goes from integers 0 to N and then the second if statement which adds the element in the arraylist.

4. Explanations for ComplexCode18:

This solution is Nlogn. This is because the tree's fundamental coding structure looks like a branch going across and multiplying in an exponential way similar to 1 to 2 to 4 to 8 to 16 to etc. So the logn n makes sense because the data structure that is used is a tree however the N in the solution of Nlogn is because of the for loop which goes from int i = 0 when i is less than N.

5. Explanations for ComplexCode20:

The reason that this code is log N is because the recursion allows the function to be repeated exponentially by so for example, it will be N/2 then N/4 then N/8 then N/16 so N is being divided by factors of 2. This means that two to some power x will be equal to N so the running type analysis is log N.