CMPT 706, Spring 2020
Quiz 2 - February 13

Name_________________________

SFU ID: |__|__|__|__|__|__|__|__|__|

Instructions:

1. Write your name and SFU ID **clearly**.
2. No calculators, no cell phones, or any other material.
3. Write your answers in the provided space.
4. Explain all your answers.

Question 1 (10 points)

Explain the Miller-Rabin Primality Test for N= 377 with a=12.
You may use the following facts:

- 376=47*8.
- 12^{47}=220 (mod 377)
- 12^{94}=144 (mod 377)
- 12^{188}=1 (mod 377)
- 12^{376}=1 (mod 377)

What is your conclusion from this execution of the test?

The test checks that 12^{376}=1 and continues next.
Then, it checks 12^{188}=1, and continues next.
Then it checks 12^{94}=144. Seeing this the algorithm outputs COMPOSITE.

Conclusion: 377 is composite. This is because 144^2=1 (mod 377). That is, 1 has a non-trivial square root, i.e., a square root that is no 1 or -1.

Question 2 (20 points)

Consider the following version of MergeSort:
Given an array of length n,

1. Split the array into 3 equal parts (rather than 2)
2. Sort each of the parts recursively
3. Merge the three parts

[10 points] Explain how to implement the merge part in O(n) time.

We will define an array RET of length n that will contain the resulting sorted array.

Each of the subarrays will have a pointer starting from the first element.
In each iteration we will compare the three pointers.
The minimum of the three will be moved to the RET, and the corresponding pointer will be incremented by 1 to the next index.
When one of the pointers reaches the end of its array, we continue with the remaining two arrays.

The runtime is clearly O(n), because in each iteration we make at most 3 comparisons and move one of the elements to RET. Therefore, the total number of iterations will be n.

[10 points] What is the runtime of the algorithm on arrays of length n?
Use big-O notation to express your answer. Explain your solution.

Denote the runtime of Merge-Sort by T(n).
The algorithm makes 3 recursive calls on subarrays of size n/.3 and calls the merging procedure, which runs in O(n) time.

Therefore, $T(n) = 3T(n/3) + O(n)$.
By applying the Master Method, we get that $T(n)=O(n \log(n))$.