**COSC 3304 – Algorithms Design and Analysis**

**Assignment 3**

**Due: 23:59:00pm, 02/06/2024**

1. (30 points) Please show the big O notation of the recurrence relation below using the substitution method (please show detailed steps):

T(n)= T(n1/2) + 3

(Base case T(2))

* T(n)= T(n1/2) + 3
  + T(n1/2)=T((n1/2)1/2) + 3 = T(n1/4) + 3
* T(n) = T(T(n1/4) + 3 ) + 3
  + T(n1/4) = T((n1/4)1/2) + 3 = T(n1/8) + 3
* T(n) = T(T(T(n1/8) + 3) +3) +3 ….
* Let us say that k is the amount of steps you need to get to the base case (T(2)) , then we can rewrite the formula as
* T(n) = T(n1/(2^k)) + 3k
  + This means that n1/(2^k) = 2, now we have to isolate k
  + logn2 = 1/(2k) by logarithmic properties
  + 1/logn(2) = 2k
    - If we say that 1 = lognn, then we can replace
    - (lognn) / (logn2) = log2n
  + log2n = 2k
    - by exponentiating both sides by 2, we can cancel out the log2
    - 2log2n = n
  + n = 22^k
  + k = log2(log2(n))
* T(n) = T(2) + 3k
* T(n) = T(2) + 3 log2(log2(n))
* The most contributing term would be the 3 log2(log2(n)) , so our big O notation would be O(log(logn))

1. (20 points) An array contains a descending subarray followed by an ascending subarray. Please write a pseudocode to find the minimum element in the array using O(logn).

* We can use a algorithm similar to that of binary search to obtain an O(logn) time complexity.
* A computer screen shot of white text

  Description automatically generated

1. (20 points) Write a pseudo code function *m(i)* to compute the following series using a recursive method.



* We will be using top down recursion
* A computer screen shot of white text

  Description automatically generated

1. (15 points) Please sort the input array [9 -9 15 35 -1 14 20 7] using the **INSERTION** algorithm (please show detailed steps for full credits).

A math equations on an orange background

Description automatically generated

A screenshot of a grid

Description automatically generated

1. (15 points) Please sort the input array [9 -9 15 35 -1 14 20 7] using the **BUBBLESORT** algorithm (please show detailed steps for full credits).

A grid of numbers and numbers

Description automatically generated