SE2228 - Analysis and Design of Algorithms: Assignment 1 DUE: 07.03.2023

ID & Name

1. a)Determine the following sum:****

Answer to 1.a): This is a geometric series where a = 1 and r = 3, so by the sum of geometric series formula we get:

b)Determine the following sum: ****

Answer to 1.b): This expression can be simplified by the following steps:

c) Determine the following sum :

Answer to 1.c): This expression can be simplified by the following steps:

2. What are the complexities of the following code segments in terms of n? Give an upper bound.

a)

int i=1;  
while (i<= n) {

int j = i;  
 while (j > 0) {  
 j = j/2;

}  
 i++;

}

Answer to 2.a): O(nlogn) since this is sum of logarithms from 1 to n, which can be expressed as O(logn!) which can be approximated to O(nlogn) but it can be simplified to O(n) since the inner loop dominates the time complexity of the outer loop.

b)

sum = 0;

for (i = 1; i < n; i++)

for (j = 1; j < i\*i; j++){

if (j % i == 0)

for (k = 0; k < j; k++)

sum++;

}

Answer to 2.b): O(n^3)

c)

while (n > 0) {

for (int i=0; i<n; i++) {

sum++;

}

n = n/2;

}

Answer to 3.c): The total time complexity of the given code is O(n\*log(n)). However, since the time complexity of the inner for loop dominates over the time complexity of the outer while loop, we can simplify the time complexity to O(n).

3.Determine complexities of the following functions (Big-O):

1. : O(nlogn)
2. : O(n^3)
3. : O(2^N)
4. : O(n^4)
5. : O(n^3)
6. : O(N)
7. : O(N!)

4.For each of the following functions, indicate how much the function’s value will change if its argument is increased **q** times. Muhamed has 7 letters so:

1. -> log

4.a)An algorithm has O(√ n) runtime. When run for n=10000,it takes 10 seconds.How long would it run for an input size of 100000? seconds

b)An algorithm has O(nlog n) runtime. When run for n=16, it takes 32 seconds.How long would it run for an input size of 64?

seconds

4.Solve the following recurrence by substitution:

T(n) = T(n-1) + n

=0 if n = 0

T(n) = T(n-1) + n = T(n-2) + (n-1) + n = T(n-3) + (n-2) + (n-1) + n = T(0) + 1 + 2 + …. + n  
since T(0) = 0, this is an arithmetic sum: T(n) = 1 + 2 + … + n = n(n+1)/2