**Assignments - Time complexity**

**Question 1. Analyze the time complexity of the following Java code and suggest a way to improve it:**

int sum = 0;

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

for(int j = 1; j <= i; j++) {

sum++;

}

}

**answer-** The time complexity of this code is O(n^2) as it uses nested loops, where the outer loop runs n times and the inner loop runs i times where i varies from 1 to n.

One way to improve the time complexity of this code is to use a mathematical formula to find the sum instead of

using nested loops.

**Question 2: Find the value of T(2) for the recurrence relation T(n) = 3T(n-1) + 12n, given that T(0) = 5.**

**answe**r=Substituting the values in the relation:

T(1) = 3T(0) + 12

=> T(1) = 15 + 12 = 27

T(2) = 3T(1) + 12 \* 2

=>T(2) = 3 \* 27 + 24 = 81 + 24

Hence T(2) = 105.

**Question 3: Given a recurrence relation, solve it using a substitution method.  Relation : T(n) = T(n - 1) + c**

**Answer**- Let the solution be T(n) = O(n), now let’s prove this using the induction method.For that to happen T(n) <= cn where c is some constant.

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

T(n - 1) = T(n - 2) + c

T(n - 2) = T(n - 3) + c

T(2) = T(1) + c

—------------------------ Adding all above equations

T(n) = T(1) + cn

Let us assume T(1) to be a constant value.

T(n) = k + cn

Therefore, T(n) <= cn

Hence we can conclude T(n) = O(n).

**Question 4: Given a recurrence relation:**

**T(n) = 16T(n/4) + n2logn**

**Find the time complexity of this relation using the master theorem**

**answer-**The relation:T(n)=16T(n/4)+n2logn

Here,a=16

b=4

k=2

p=1

bk=42=16

Here a=bk

Also p>-1

Hence T(n)=θ(nlogab\*logp+1n)

Therefore T(n)=θ(nlog164\*log1+1n)=θ(n1/2log2n)

**—-------------------------------------------------------------------------------------------------------------------------**

.