# Time and Space complexity

# Assignment Questions

## Assignment Questions

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++;
   }
   }
   ```

   **Ans :**

   **The time complexity of the given code is O(n^2) because there is the nested loop**

   **We can reduce the time complexity by writing the simple formula to calculate the sum of natural number from 1 to n-1**

   **Sum = n (n-1) / 2**

   **So the time complexity will be O(1)**

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

   **Ans :**

   **By using substitution method we solve it**

   **T(n) = 3 T(n-1) + 12n — eq 1**

**Lets find the value of T(n-1) by n to n-1 in the T(n) equation**

**T(n-1) = 3 T(n-2) + 12(n-1)**

**T(n-1) = 3 T(n-2) + 12n - 12**

**Put the equation of T(n-1) in the eq 1**

**T(n) = 3 (3 T(n–2) + 12n - 12 ) + 12n**

**T(n) = 9 T(n-2) + 36n - 36 + 12n**

**T(n) = 9 T(n-2) + 48n - 36 — eq 2**

**Put n = 2 in eq 2 for finding T(2)**

**T(2) = 9 T(2-2) + 48 * 2 -36**

**T(2) = 9 T(0) + 48 * 2 -36**

**T(2) = 9 * 5 + 48 * 2 - 36**

**T(2) = 105**

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

**Ans :**

**Relation : T(n) = T(n-1) + C — eq 1**

**Put n = n-1 for finding T(n-1)**

**T(n-1) = T (n-2) + C**

**Put T(n-1) in eq 1**

**T(n) = T(n-2) + 2C — eq 2**

**T(n-2) = T(n-3) + C**

**Put in eq 2**

**T(n) = T(n-3) + 3C — eq 3**

**T(n) = T(n-k) + KC**

**n - k = 1**
**k = n-1**

**T(1) = 0**

**T(n) = T(n-n+1) + Cn - C**

**T(n) = T(1) + Cn - C**

**T(n) = Cn - C**

**Therefore,**

**Time complexity = O(n)**

**4. Given a recurrence relation: T(n) = 16T(n/4) + n2logn Find the time complexity of this relation using the master theorem.**

**Ans :**

**Comparing with the standard equation of masters theorem**

**T(n) = a T(n/b) + O(n^k log n ^p+1)**

**By comparing we get**

**a = 16 , b = 4 , k = 2 , p = 1**

**b^k = 4^2 = 16**

**a = b^k**

**T(n) = O(n^loga base b log n ^ p+1)**

**T(n) = O(n^2 log n ^2)**

**Time complexity = O(n^2 log n ^ 2)**

**5. Solve the following recurrence relation using recursion tree method T(n) = 2T(n / 2) + n**

**Ans :**

**T(n) = T(n/2) + T(n/2) + n**

**We get the series of the elements**

**( n + n + n + ……. K times )**

**n ( 1 + 1 + 1 + ….. K times )**

**The value of k becomes log n**

**Therefore,**

**The Time complexity is O(n log n)**

**6. T(n) = 2T(n / 2) + K Solve using Recurrence tree method.**

**Ans :**

**T(n) = T(n/2) + T(n/2) + k**

**K will be equal to log n**

**( K + 2K + 4K )**

**T(n) = ( K + 2K + 3K + 4K + … ) + O(n)**

**T(n) = K + O(n)**

**Time complexity = O(n)**