**Assignment:1**

**Question 1. Analyse 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:**

Time complexity of this program to find the sum of n numbers is : BigO(n).

We can make it BigO(1) is by using the mathematical formula to find the sum of n natural number.

Code:

public class Main

{

public static void main(String[] args) {

int n=10;

int sum = 0;

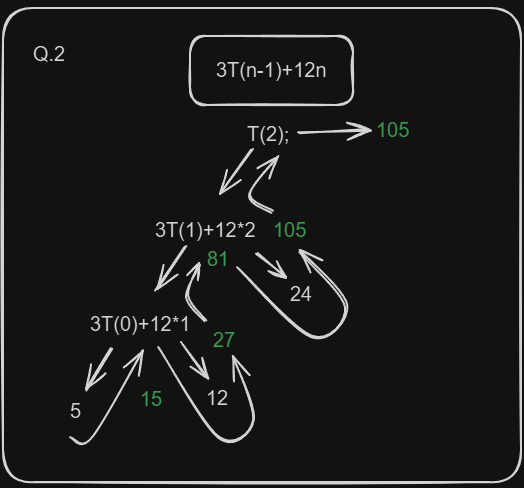
sum=n\*(n+1)/2;

System.out.print(sum);

}

}

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

Ans:

**Assignment 2:**

**Question 1: Take m and n input from the user and m \* n integer inputs from user and print the following: number of positive numbers number of negative numbers number of odd numbers number of even numbers number of 0.**

**Ans:** [**code**](https://gist.github.com/Prince-GH/20542191a0a7af46f79b023970012adb)

**Question 2: write a program to print the elements above the secondary diagonal in a user inputted square matrix.**

**Ans:** [**code**](https://gist.github.com/Prince-GH/d9d924c68769c3504672c87a21a27794)

**Question 3: write a program to print the elements of both the diagonals in a user inputted square matrix in any order.**

**Ans:** [**code**](https://gist.github.com/Prince-GH/7833d520572c1067bb56affaf37b678a)

**Question 4: Write a program to find the largest element of a given 2D array of integers.**

**Ans:** [**code**](https://gist.github.com/Prince-GH/c77cc850f9bbefd02eac7290c75555d7)

**Question 5: Write a function which accepts a 2D array of integers and its size as arguments and displays the elements of middle row and the elements of middle column. Printing can be done in any order. [Assuming the 2D Array to be a square matrix with odd dimensions i.e. 3x3, 5x5, 7x7 etc...].**

**Ans:** [**code**](https://gist.github.com/Prince-GH/6c3a7e81cfb4a63273cb1dbf6a888440)