21BCE7371

RADHA KRISHNA GARG

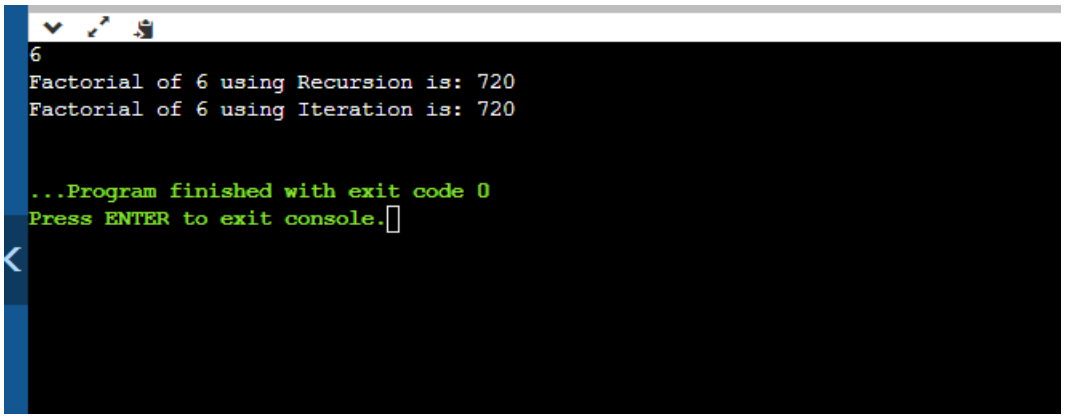
DSA LAB ASSIGNMENT-1

1. Write a Program to find the factorial of a given number using recursion and analyze the time complexity.

INPUT

```java
// Java program to find factorial of given number
import java.util.*;
class Main {

    // ----- Recursion -----
    // method to find factorial of given number
    static int factorialUsingRecursion(int n)
    {
        if (n == 0)
            return 1;

        // recursion call
        return n * factorialUsingRecursion(n - 1);
    }

    // ----- Iteration -----
    // Method to find the factorial of a given number
    static int factorialUsingIteration(int n)
    {
        int fact = 1, i;

        // using iteration
        for (i = 2; i <= n; i++)
            fact *= i;

        return fact;
    }

    // Driver method
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int num = sc.nextInt();
        //int num = 5;
        System.out.println("Factorial of " + num
                           + " using Recursion is: "
                           + factorialUsingRecursion(num));


        System.out.println("Factorial of " + num
                           + " using Iteration is: "
                           + factorialUsingIteration(num));
    }
}
```

OUTPUT




```
// Java program to find factorial of given number
import java.util.*;
class Main {

    // ----- Recursion -----
    // method to find factorial of given number
    static int factorialUsingRecursion(int n)
    {
        if (n == 0)
            return 1;

        // recursion call
        return n * factorialUsingRecursion(n - 1);
    }

    // ----- Iteration -----
    // Method to find the factorial of a given number
    static int factorialUsingIteration(int n)
    {
        int fact = 1, i;
```

```
            // using iteration
            for (i = 2; i <= n; i++)
                    fact *= i;

            return fact;
        }

        // Driver method
        public static void main(String[] args)
        {
           Scanner sc = new Scanner(System.in);
           int num = sc.nextInt();
            //int num = 5;
            System.out.println("Factorial of " + num
                                        + " using Recursion is: "
                                        + factorialUsingRecursion(num));



            System.out.println("Factorial of " + num
                                        + " using Iteration is: "
                                        + factorialUsingIteration(num));
        }
}
```

TIME COMPLEXITY

T(n) = T(n-1) + 3 (3 is for as we must do three constant operations like multiplication, and checking the value of n in each recursive call)

= T(n-2) + 6 (Second recursive call)
= T(n-3) + 9 (Third recursive call)
.
.
.
.
= T(n-k) + 3k
till, k = n

Then,

= T(n-n) + 3n
= T (0) + 3n
= 1 + 3n

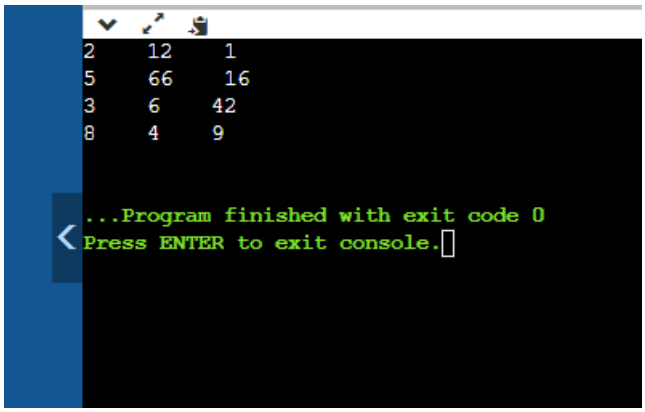
The time complexity of recursive factorial is O(n)

Write a Program to find the transpose of a given matrix and display its time complexity.
Challenging

INPUT

```java
class Main
{
    public static void main (String[] args)
    {
        // given matrix M whose transpose is to be found
        int M[][] = {{2,5,3,8},
                     {12,66,6,4},
                     {1,16,42,9}};

        // find number of rows and columns in matrix M
        int n = M.length, m = M[0].length;

        //create empty transpose matrix of size m*n
        int M_transpose[][] = new int[m][n];

        //traverse matrix M
        for(int i=0;i<n;i++){
            for(int j=0;j<m;j++){

                //assign M_transpose[j][i] as M[i][j]
                M_transpose[j][i] = M[i][j];
            }
        }

        // output the transpose matrix
        for(int i=0;i<m;i++){
            for(int j=0;j<n;j++){
                System.out.print(M_transpose[i][j] + "    ");
            }
            System.out.println();
        }

    }
}
```

OUTPUT




```
class Main
{
        public static void main (String[] args)
        {
          // given matrix M whose transpose is to be found
                int M[][] = {{2,5,3,8},
                             {12,66,6,4},
                             {1,16,42,9}};

    // find number of rows and columns in matrix M
    int n = M.length, m = M[0].length;

    //create empty transpose matrix of size m*n
    int M_transpose[][] = new int[m][n];

    //traverse matrix M
    for(int i=0;i<n;i++){
      for(int j=0;j<m;j++){

        //assign M_transpose[j][i] as M[i][j]
        M_transpose[j][i] = M[i][j];
      }
    }

    // output the transpose matrix
    for(int i=0;i<m;i++){
      for(int j=0;j<n;j++){
        System.out.print(M_transpose[i][j] + "    ");
      }
      System.out.println();
    }
```

```
  }
}
```

TIME COMPLEXITY

Time complexity is O(n*m)

N – number of rows

M—number of columns

3 Write a Program to illustrate the difference between recursion and iteration by giving its time complexities.

```java
// Java program to find factorial of given number
import java.util.*;
class Main {

    // ----- Recursion -----
    // method to find factorial of given number
    static int factorialUsingRecursion(int n)
    {
        if (n == 0)
            return 1;

        // recursion call
        return n * factorialUsingRecursion(n - 1);
    }

    // ----- Iteration -----
    // Method to find the factorial of a given number
    static int factorialUsingIteration(int n)
    {
        int fact = 1, i;

        // using iteration
        for (i = 2; i <= n; i++)
            fact *= i;

        return fact;
    }

    // Driver method
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int num = sc.nextInt();
        //int num = 5;
        System.out.println("Factorial of " + num
                           + " using Recursion is: "
                           + factorialUsingRecursion(num));


        System.out.println("Factorial of " + num
                           + " using Iteration is: "
                           + factorialUsingIteration(num));
    }
}
```

OUTPUT

```
6
Factorial of 6 using Recursion is: 720
Factorial of 6 using Iteration is: 720


...Program finished with exit code 0
Press ENTER to exit console.
```

```
// Java program to find factorial of given number
import java.util.*;
class Main {

	// ----- Recursion -----
	// method to find factorial of given number
	static int factorialUsingRecursion(int n)
	{
		if (n == 0)
			return 1;

		// recursion call
		return n * factorialUsingRecursion(n - 1);
	}

	// ----- Iteration -----
	// Method to find the factorial of a given number
	static int factorialUsingIteration(int n)
	{
		int fact = 1, i;

		// using iteration
		for (i = 2; i <= n; i++)
			fact *= i;

		return fact;
	}

	// Driver method
```

```
        public static void main(String[] args)
        {
            Scanner sc = new Scanner(System.in);
            int num = sc.nextInt();
                //int num = 5;
                System.out.println("Factorial of " + num
                                                + " using Recursion is: "
                                                + factorialUsingRecursion(num));



                System.out.println("Factorial of " + num
                                                + " using Iteration is: "
                                                + factorialUsingIteration(num));
        }
}
```

TIME COMPLEXITY

T(n) = T(n-1) + 3 (3 is for as we must do three constant operations like
multiplication, and checking the value of n in each recursive
call)

= T(n-2) + 6 (Second recursive call)
= T(n-3) + 9 (Third recursive call)

.
.
.
.
$= T(n-k) + 3k$
till, $k = n$

Then,

$= T(n-n) + 3n$
$= T(0) + 3n$
$= 1 + 3n$

The time complexity of recursive factorial is O(n)