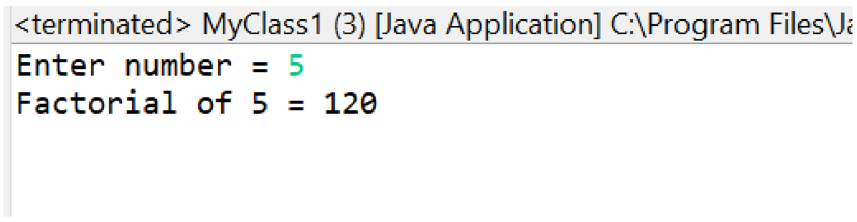
Data Structures & Algorithms (CSE2001) Lab-1 03-09-2022

KHAN MOHD OWAIS RAZA (20BCD7138)

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

User-Input Java Code and Output :-

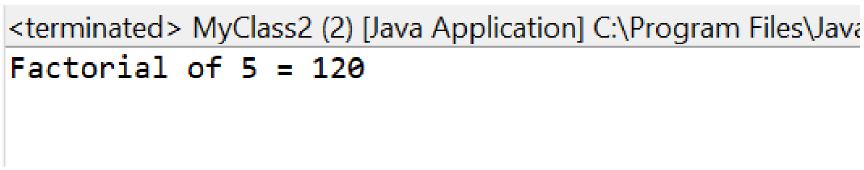
```java
/**
 Name: KHAN MOHD OWAIS RAZA
 ID : 20BCD7138
 Course: Data Structures & Algorithm
 Code: CSE2001
 Slot: L19+L20
**/
/* Java code to find factorial of number using recursion - User
Input code*/
package CSE2001_Lab1_20BCD7138;
import java.util.Scanner;
public class MyClass1{
static int factorial(int X){
if (X == 0) return 1;
else return(X*factorial(X-1));
}
public static void main(String args[]){
int i, FACTORIAL, NUMBER;
Scanner s = new Scanner(System.in);
System.out.print("Enter number = ");
NUMBER = s.nextInt();
FACTORIAL = factorial(NUMBER);
System.out.println("Factorial of "+NUMBER+" = "+FACTORIAL);
}}
```



```
<terminated> MyClass1 (3) [Java Application] C:\Program Files\J
Enter number = 5
Factorial of 5 = 120
```

Java Code and Output :-

```
/**
 Name: KHAN MOHD OWAIS RAZA
 ID : 20BCD7138
 Course: Data Structures & Algorithm
 Code: CSE2001
 Slot: L19+L20
**/
/* Java code to find factorial of number using recursion - Without
User Input*/
package CSE2001_Lab1_20BCD7138;
public class MyClass2{
static int factorial(int X){
if (X == 0) return 1;
else return(X * factorial(X-1));
}
public static void main(String args[]){
int i,FACTORIAL=1;
int NUMBER = 5;
FACTORIAL = factorial(NUMBER);
System.out.println("Factorial of "+NUMBER+" = "+FACTORIAL);
}}
```



```
<terminated> MyClass2 (2) [Java Application] C:\Program Files\Java
Factorial of 5 = 120
```

Time complexity :-

We have :

```
factorial(int X){
if (X == 0) return 1;
else return(X * factorial(X-1));
}
```

Thus time complexity for single recursive call is :

T(X) = T(X-1) + 3   (first recursive call)

T(X) = T(X-2) + 6   (second recursive call)

T(X) = T(X-3) + 9   (third recursive call)

…..

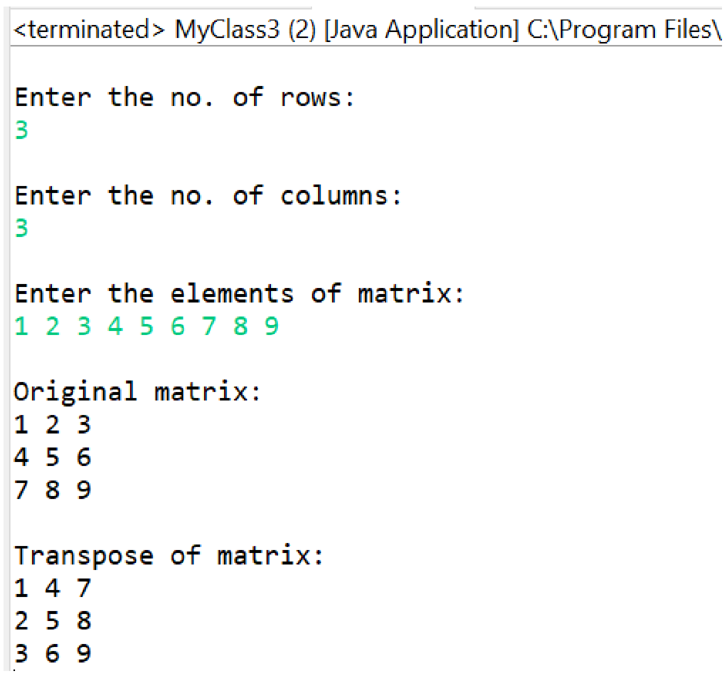
T(X) = T(X-k) + 3*k (Till k=X)

Thus, T(X) = T(X-X) + 3*X = T(0) + 3*X = 1 + 3*X

Hence, time complexity is O(X)

2. Write a program to find the transpose of a given matrix and display its time complexity

```java
/**
 Name: KHAN MOHD OWAIS RAZA
 ID : 20BCD7138
 Course: Data Structures & Algorithm
 Code: CSE2001
 Slot: L19+L20
**/
/* Java code to find a transpose of a matrix */
package CSE2001_Lab1_20BCD7138;
import java.util.*;
public class MyClass3{
public static void main(String []args){
Scanner sc=new Scanner(System.in);
int m,n;
System.out.println("\nEnter the no. of rows: ");
m=sc.nextInt();
System.out.println("\nEnter the no. of columns: ");
n=sc.nextInt();
int arr[][]=new int[10][10];
System.out.println("\nEnter the elements of matrix: ");
for(int i=0;i<m;i++){
for(int j=0;j<n;j++){
arr[i][j]=sc.nextInt();
}}
System.out.println("\nOriginal matrix: ");
for(int i=0;i<m;i++){
for(int j=0;j<n;j++){
System.out.print(arr[i][j]+" ");
}
System.out.println("");
}
int brr[][]=new int[10][10];
for(int i=0;i<m;i++){
for(int j=0;j<n;j++){
brr[j][i]=arr[i][j];
}}
System.out.println("\nTranspose of matrix: ");
for(int i=0;i<m;i++){
for(int j=0;j<n;j++)
{
System.out.print(brr[i][j]+" ");
}
System.out.println("");
}}}
```



```
<terminated> MyClass3 (2) [Java Application] C:\Program Files\

Enter the no. of rows:
3

Enter the no. of columns:
3

Enter the elements of matrix:
1 2 3 4 5 6 7 8 9

Original matrix:
1 2 3
4 5 6
7 8 9

Transpose of matrix:
1 4 7
2 5 8
3 6 9
```

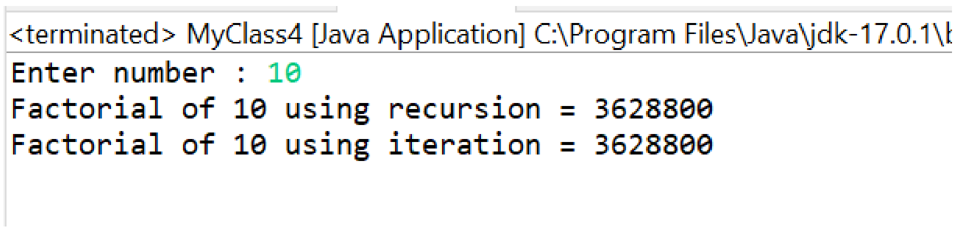
Time complexity :-

In the above code, we execute two loops i.e. first loop R1 times and second loop C1 times.
Let n = maximum no. of R1 and C1.

Thus, time complexity = O(n*n) = O(square(n)).

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

```
/**
 Name: KHAN MOHD OWAIS RAZA
 ID : 20BCD7138
 Course: Data Structures & Algorithm
 Code: CSE2001
 Slot: L19+L20
**/
package CSE2001_Lab1_20BCD7138;
import java.util.Scanner;
public class MyClass4 {
static int factorialUsingRecursion(int n){
if (n == 0) return 1;
return n*factorialUsingRecursion(n-1);
}
static int factorialUsingIteration(int n){
int res = 1, i;
for (i = 2; i <= n; i++) res *= i;
return res;
}
public static void main(String[] args){
int num;
Scanner s = new Scanner(System.in);
System.out.print("Enter number : ");
num = s.nextInt();
System.out.println("Factorial of " + num + " using recursion = " +
factorialUsingRecursion(num));
System.out.println("Factorial of " + num+ " using iteration = " +
factorialUsingIteration(num));
}}
```



```
<terminated> MyClass4 [Java Application] C:\Program Files\Java\jdk-17.0.1\
Enter number : 10
Factorial of 10 using recursion = 3628800
Factorial of 10 using iteration = 3628800
```

Time complexity :-

For recursion :

T(n) = T(n-1) + 3   (first recursive call)

T(n) = T(n-2) + 6   (second recursive call)

T(n) = T(n-3) + 9   (third recursive call)

…………

T(n) = T(n-k) + 3*k   (till n=k)

Thus, T(n) = T(n-n) + 3*n = T(0) + 3*n = 1 + 3*n

Hence, time complexity = O(n)

For iteration :

Time complexity = O(n^k) where k= no. of nested loops

Thus for single nested loop, it is O(n)

And for no nested loop, it is O(1)