PRACTICAL – 1

Name:- Manish M Prajapati (190420116054)

Subject:- Analysis and Design of Algorithms (3150703)

Class:- IT 3rd Year, 5th semester.

Question:- Implementation and Time analysis of factorial program using iterative and recursive method.

Answer:-

```
#include<stdio.h>
#include<conio.h>

//PRACTICAL_1
//Implementation and Time analysis of factorial program using iterative and recursive method


int count1=0, count2=0;

int factorial(int n)
{
	count1++;
	if(n==0 || n==1)
	{
		return 1;
	}
	else if(n==2)
	{
		return 2;
	}
	else
	{
		return n*factorial(n-1);
	}
}

int main()
{
	int i, number, fact1=1, fact2=1;
	printf("Enter a number you want to find factorial of: ");
	scanf("%d", &number);
	printf("\n");

	for(i=1; i<=number; i++)
	{
		count2++;
		fact1 *=i;
	}
```

```
    printf("The time complexity of this program through both the methods is O(n).\n\n");

    printf("The factorial of number %d through iterative method is %d.", number, fact1);
    printf("\nThe time taken is %d number of iterations.\n\n", count2);

    fact2 = factorial(number);

    printf("The factorial of number %d through recursive method is %d.", number, fact2);
    printf("\nThe time taken is %d number of iterations.", count1);

    return 0;
}
```

Output:-

```
Enter a number you want to find factorial of: 5

The time complexity of this program through both the methods is O(n).

The factorial of number 5 through iterative method is 120.
The time taken is 5 number of iterations.

The factorial of number 5 through recursive method is 120.
The time taken is 4 number of iterations.
```

TIME ANALYSIS FOR RECURSIVE METHOD

Worst Case:-

The function factorial is depended on the value of n. Each time the function runs, there will be only 1 statement executed in the function and the value will be returned back.

Every time the function is called, the statements executing will be only one.
So, for a number n, n!= n*(n-1)*(n-2)*(n-3)*....*3*2*1

Hence, total required statements would be;
S(n) = 1 + 1 + 1 + 1 + 1.....n-1 times..(because the function returns 2 directly when n=2)
S(n) = 1*(n-1)
S(n) = n-1
S(n) ≈ n .

Therefore, the time complexity of worst case recursive factorial method is O(n).

Average Case:-

As the function value is depended on the value n, that, the number of times the function would be recalled, so, at lower values of n, the time analysis becomes quick.

Let us consider that we want to find the factorial of 5. So, the value 5 will be passed to factorial function. The value, at each pass of function will get decremented, i.e., in order, 5, 4, 3, 2. In first pass, only 1 statement is executed, i.e., the last one. In that phase, again the function is recalled with one value less than 5, i.e., with 4. This will continue till n becomes 1.

So, the iterations are:
S(n) = 1 + 1 + 1 + 1 (one iteration is less because of direct returning the value 2 for n=2)
S(n) = 4
S(n) = n-1 ≈ n

Hence, for average case time complexity of recursive method for factorial is O(n).

Best Case:-

In this program, the best case scenario is for value: 0, 1 & 2.

Let us consider that value 2 is entered. Passing that value to the function, checking the condition it's factorial 2 is returned with no further iteration.

So, the iteration is only: 1

Hence, the time complexity for best case recursive method factorial is O(1).

--------_________--------_________--------

TIME ANALYSIS FOR ITERATIVE METHOD

Worst Case:-

The part of program dedicated to calculate the factorial is wholly depended on the value of n. The number of iterations required would be equal to n.

Let us consider for greater value of n. In that case, every time the loop executes, the value of n will decrease, i.e., n*(n-1)*(n-2)*.....*3*2*1. Every time the loop executes, there will be only 1 iteration required.

So, the number of iterations would be;
S(n) = 1 + 1 + 1 + 1 + ... + n times
S(n) = 1*n
S(n) = n .

Hence, the time complexity of worst case iterative factorial method is O(n).

Average Case:-

As the number of iterations is depended on the value of ‘n’, hence, for average case, it becomes easy to calculate the time complexity.

Let us consider, the value of n = 4, in that case, the value will do in decreasing as: 4, 3, 2, 1.

So, the iterations required will be:
S(n) = 1 + 1 + 1 + 1
S(n) = 4
S(n) = n .

Hence, from above case, the time complexity for average case iterative factorial method is O(n).

Best Case:-

In this program for value of n : 0 & 1. There will be only 1 iteration in ‘for’ loop.

So, the total iterations would be: S(n) = 1.

Hence, for best case iterative factorial method, the time complexity is O(1).