

Design and Analysis of Algorithms

-Space Complexity

Name: J Darunkumar

Roll-No: CH.SC.U4CSE24111

1. Write a program to find sum of first n natural numbers, using user defined function.

Code:

```
1 #include <stdio.h>
2
3 int add(int n){
4     int sum=0;
5     for(int i=1;i<n+1;i++){
6         sum+=i;
7     }
8     return sum;
9 }
10
11 int main(){
12     int n;
13     printf("Enter a number to find the sum up to it:");
14     scanf("%d",&n);
15     printf("The sum is:%d\n",add(n));
16     return 0;
17 }
```

Output:

```
[darunkumar@DK] - [~/Downloads]
$ gcc sumn.c

[darunkumar@DK] - [~/Downloads]
$ ./a.out
Enter a number to find the sum up to it:10
The sum is:55
```

Space complexity:

The space used by the above program remains constant since it has no variable components. Therefore, the space complexity is O(1).

2. Write a program to find sum of squares of first n natural numbers.

Code:

```
1 #include <stdio.h>
2
3 int main(){
4     int n,sum=0;
5     printf("Enter a natural number to find the sum of squares:");
6     scanf("%d",&n);
7     for(int i=1;i<n+1;i++){
8         sum+=i*i;
9     }
10    printf("The sum of squares is %d\n",sum);
11    return 0;
12 }
```

Output:

```
[darunkumar@DK]~/Downloads]$ gcc squaresSum.c
[darunkumar@DK]~/Downloads]$ ./a.out
Enter a natural number to find the sum of squares:10
The sum of squares is 385
```

Space Complexity:

The space used by the above program remains constant since it has no variable components. Therefore, the space complexity is O(1).

3. Write a program to find sum of cubes of first n natural numbers.

Code:

```
1 #include <stdio.h>
2
3 int main(){
4     int n,sum=0;
5     printf("Enter a natural number to find the sum of cubes:");
6     scanf("%d",&n);
7     for(int i=1;i<n+1;i++){
8         sum+=i*i*i;
9     }
10    printf("The sum of cubes is %d\n",sum);
11    return 0;
12 }
```

Output:

```
[~] darunkumar@DK ~]$ gcc cubeSum.c
[~] darunkumar@DK ~]$ ./a.out
Enter a natural number to find the sum of cubes:10
The sum of cubes is 3025
```

Space Complexity:

The space used by the above program remains constant since it has no variable components. Therefore, the space complexity is O(1).

4. Write a program to find a factorial of the given integer using recursion.

Code:

```
1 #include <stdio.h>
2
3 int fac(int n){
4     if(n==0)
5         return 1;
6     return n*fac(n-1);
7 }
8
9 int main(){
10    int n;
11    printf("Enter a number to find it's factorial:");
12    scanf("%d",&n);
13    printf("The factorial of %d is %d\n",n,fac(n));
14    return 0;
15 }
```

Output:

```
[darunkumar@DK] - [~/Downloads]
$ gcc factorial.c

[darunkumar@DK] - [~/Downloads]
$ ./a.out
Enter a number to find it's factorial:10
The factorial of 10 is 3628800
```

Space complexity:

The recursive function is the only part of the program that grows with the input, so the space complexity is $O(n)$. The total space depends entirely on the input size.

5. Write a program for transposing a 3X3 matrix.

Code:

```
1 #include <stdio.h>
2
3 int main(){
4     int arr[3][3];
5     printf("-----Enter the elements of the 3x3 matrix-----\n");
6     for(int i=0;i<3;i++){
7         for(int j=0;j<3;j++){
8             printf("Enter element %d %d:",i+1,j+1);
9             scanf("%d",&arr[i][j]);
10        }
11    }
12    printf("-----The initial matrix-----\n");
13    for(int i=0;i<3;i++){
14        for(int j=0;j<3;j++){
15            printf("%d ",arr[i][j]);
16        }
17        printf("\n");
18    }
19    int tarr[3][3]={0};
20    for(int i=0;i<3;i++){
21        for(int j=0;j<3;j++){
22            tarr[i][j]=arr[j][i];
23        }
24    }
25    printf("-----The matrix after transpose-----\n");
26    for(int i=0;i<3;i++){
27        for(int j=0;j<3;j++){
28            printf("%d ",tarr[i][j]);
29        }
30        printf("\n");
31    }
32    return 0;
33 }
```

Output:

```
[darunkumar@DK:~/Downloads]$ gcc matrix.c  
[darunkumar@DK:~/Downloads]$ ./a.out  
-----Enter the elements of the 3x3 matrix-----  
Enter element 1 1:1  
Enter element 1 2:2  
Enter element 1 3:3  
Enter element 2 1:4  
Enter element 2 2:5  
Enter element 2 3:6  
Enter element 3 1:7  
Enter element 3 2:8  
Enter element 3 3:9  
-----The initial matrix-----  
1 2 3  
4 5 6  
7 8 9  
-----The matrix after transpose-----  
1 4 7  
2 5 8  
3 6 9
```

Space Complexity:

The size of the matrix is fixed in the above program so the space remains constant, the space complexity is O(1)

6. Write a program to find Fibonacci series.

Code:

```
1 #include <stdio.h>
2
3 int main(){
4     int n,first=0,second=1;
5     printf("Enter the number of fibonacci terms to print:");
6     scanf("%d",&n);
7     for(int i=0;i<n;i++){
8         printf("%d ",first);
9         int temp=second;
10        second=first;
11        first+=temp;
12    }
13    printf("\n");
14    return 0;
15 }
```

Output:

```
[darunkumar@DK] - [~/Downloads]
$ gcc fibo.c

[darunkumar@DK] - [~/Downloads]
$ ./a.out
Enter the number of fibonacci terms to print:10
0 1 1 2 3 5 8 13 21 34
```

Space Complexity:

The program does not have any variable part of space, so the space remains constant and the space complexity is O(1).