Design and Analysis of Algorithms

Nikesh Kumar

3rd January 2020

Contents

1 Introduction to Algorithms

An algorithm is a sequence of unambiguous instructions for solving a problem i.e. for obtaining a required output for any legitimate(valid) input in finite time.

1.1 Performance/Analysis of Algorithms

It refers to the memory and time representation of the program.

Methods of Analysis:

- Analytical
- Experimental

Any algorithm is analysed on the following criteria:

- Space Complexity
- Time Complexity

1.1.1 Space Complexity

It is the amount of memory required for a program to completion. It has 3 categories:

- Instruction Space (Compiled Program)
- Data Space (Space needed by var/const)
- Environment Stack Space (Recursive calls)

Denoted by: $C + S_p$

Sample Questions

1. Sum of array without recursion

```
int sum(int a[],int n)                                    1
{                                                         2
    int sum = 0;                                          3
    for(int i = 0;i < n;i++)                              4
        sum = sum + a[i];                                 5
    return sum;                                           6
}                                                         7
```

Space Complexity: $6x$ bytes[a]

Reason: Line 1 occupies x bytes for pointer a and x bytes for integer n. Line 3 occupies x bytes for sum and x bytes for allocating 0. Line 4 will occupy x bytes for allocating integer i. In Line 6 space will reserved for returning data.

[a] where x is bytes occupied by int

2. Sum of array with recursion

```
int sum(int a[],int n)                                    1
{                                                         2
    if(n > 0)                                             3
        return sum(a,n-1) + a[n-1];                       4
    return 0;                                             5
}                                                         6
```

Space Complexity: $3x \times (n+1)$ bytes[a]

Reason: Line 1 occupies x bytes for pointer a and x bytes for integer n. Line 4 will execute n times and each time space is reserved for pointer a and $n-1$ thus giving $3x \times n$. During the last case of $n = 0$, Line 5 will be executed returing 0 thus occupying x bytes.

[a] where x is bytes occupied by int and n is the size of array