COURSE NAME: ALGORITHMS
COURSE CODE: CIS 212

SYED TANGIM PASHA

LECTURER,

DEPARTMENT OF COMPUTING AND INFORMATION SYSTEM (CIS)

DAFFODIL INTERNATIONAL UNIVERSITY (DIU)

DHAKA,BANGLADESH

Asymptotic Analysis

- Asymptotic Analysis is a idea where we analyzing algorithms, performance of an algorithm in terms of input size (we don't measure the actual running time). We calculate, how the time (or space) taken by an algorithm increases with the input size.

- Using asymptotic analysis, we can very well conclude the best case, average case, and worst case scenario of an algorithm.
- **Best Case:** Minimum time required for program execution.
- **Average Case:** Average time required for program execution.
- **Worst Case:** Maximum time required for program execution.

Worst Case Analysis

- In worst-case analysis, we calculate the upper bound or maximum running time of an algorithm. Ex: for Linear Search the worst case happens when the element to be searched but not present in the array.
- int arr[] = {1,10,30,15), search: int x = 5
- So, the Time Complexity of Linear Search would be O(n).

- Time Complexity: Time complexity of an algorithm signifies the total time required by the program to run till its completion.

```
for ( ; ; )
{
printf( “DIU”);
}
```

Average Case Analysis

- In average case analysis, we take all the possible inputs and calculate computing time for all of the inputs. Ex: for Linear Search problem, let us assume that all cases are uniformly distributed (including the case of x not being present in the array). So, we sum all the cases and divide the sum by (n+1).

$$\text{Average Case Time} = \frac{\sum_{i=1}^{n+1} \theta(i)}{(n+1)}$$

$$= \frac{\theta((n+1)*(n+2)/2)}{(n+1)}$$

$$= \Theta(n)$$

Best Case Analysis

- In the best case analysis, we calculate lower bound on running time of an algorithm. In the linear search problem, the best case occurs when x is present at the first location.

- int arr[] = {1,10,30,15), search: int x = 1
- So, the Time Complexity of Linear Search would be O(1).

Asymptotic Analysis

- Big-oh (O) – Upper Bound
- Big-Omega (Ω) – Lower Bound
- Theta (θ) – Average Bound

Space Complexity

- Auxiliary Space: Auxiliary space is the extra space or temporary space used by an algorithm.

- Space Complexity: Space complexity of an algorithm is the space taken by the algorithm with respect to the input size. Space complexity includes both auxiliary space and space used by input.

- Space Complexity = Auxiliary Space + Input space

Different Time Complexity Graph