TECHNO INDIA UNIVERSITY
WESTBENGAL

Data Structures & Algorithms
(PCC-CS 301)

Dr. Debashis Das
Associate Professor
Department of CSE
Techno India University, Kolkata

Topics Covered

1. Complexity analysis
2. Cases of analysis

Complexity Analysis

- Complexity measure:

  - The best parameter to measure algorithm performance or complexity is its input size
    - As this parameter can estimate the performance of an algorithm regardless of the implementation style and underlying machine architecture

  - To measure an algorithm's complexity, we assume
    - A singular algorithmic step/instruction (code syntax) requires a single unit of time to execute

Complexity Analysis

- Complexity measure: time complexity

  - Assignment

    X := 10 [takes 1 unit of time (or constant time)]

  - Simple statement

    Print X [takes 1 unit of time (or constant time)]

  - If-else

```
If X = 10
  then print X
Else
   print “wrong input”
```

[takes 1 unit of time (or constant time)]

Complexity Analysis

- Complexity measure: time complexity
  - Loop

```
for i=0 to n
    statement
```

[iterates (n+1) times, takes (n+1)units of time]

If loop progress not mentioned consider it as incrementing by 1 i.e. i=i+1

  - Nested loop

Dependent Loop	Independent Loop

for i=1 to n
  for j=1 to i
  statement
 	
Time::
1+2+3+ … + n = n(n+1)/2 units
	for i=1 to n
	  for j=1 to m
	  statement
	
	Time::
	(n*m) units

Complexity Analysis

- Complexity measure: time complexity
  - Loop variants

```
For  i=1 to n
   statement
    i=i*2
```

Time::
In each step the 'i' value becomes double. The values of 'i' in consecutive steps are 1, 2, 4, 8, … (or 2^0, 2^1, 2^2, 2^3 …)

Let us assume, the loop iterates 'k' times. Hence, in k^{th} step 'i' will be 2^k which is basically 'n'

$2^k = n \Rightarrow \log_2(2^k) = \log_2(n) \Rightarrow k*\log_2(2) = \log_2(n)$ [we know, $\log_2 2 = 1$]

$k = \log_2 n$ [loop will be iterating $\log_2 n$ times, hence requires $\log_2 n$ unit of time]

Complexity Analysis

- Types of analysis
  - Best case
    - For the inputs, a problem is solved in lowest time
    - The inputs are in the form of desired output
  - Worst case
    - For the inputs, a problem is solved in longest time
    - The inputs are in complete opposite form compared to desired output
  - Average case
    - It provides a prediction about the running time of algorithm
    - The inputs are considered as random data

Complexity Analysis

- Types of analysis: example

  - Problem: sort a data set in ascending order
    - Best case

Input: 10 | 20 | 30 | 40 | 50 Output: 10 | 20 | 30 | 40 | 50

    - Worst case

Input: 50 | 40 | 30 | 20 | 10 Output: 10 | 20 | 30 | 40 | 50

    - Average case

Input: 30 | 20 | 10 | 40 | 50 Output: 10 | 20 | 30 | 40 | 50

Queries?