UNIT I INTRODUCTION:

Introduction to Algorithms – Fundamentals of Algorithmic Problem Solving – Fundamentals of the Analysis of Algorithmic Efficiency – Analysis Framework – Asymptotic Notations and Basic Efficiency Classes – Mathematical Analysis of Recursive Algorithms – Mathematical Analysis of Non-recursive Algorithms

Fundamentals of the Analysis of Algorithmic Efficiency

Fundamentals of the Analysis of Algorithm Efficiency

Issues:

- correctness
- time efficiency
- space efficiency
- optimality

Approaches:

- theoretical analysis
- empirical analysis

Total Correctness of Algorithm

- correct input data is the data which satisfies the initial condition of the specification
- correct output data is the data which satisfies the final condition of the specification

Definition

An algorithm Is called totally correct for the given specification if and only if for any correct input data it:

1 stops and
2 returns correct output

Partial Correctness of Algorithm

Usually, while checking the correctness of an algorithm it is easier to separately:

1. first check whether the algorithm stops
2. then checking the "remaining part". This "remaining part" of correctness is called Partial Correctness of algorithm

Definition: An algorithm is partially correct if satisfies the following condition: If the algorithm receiving correct input data stops then its result is correct

Note: Partial correctness does not make the algorithm stop.

An example of partially correct algorithm (computing the sum of array of numbers)

```
sum(array, len)
{
sum = 0
i = 0
while(i < len)
sum += array[i]
return sum
}
```

Is this algorithm partially correct?
Is it also totally correct?

two kinds of efficiency: time efficiency and space efficiency.

- ***Time efficiency***, also called ***time complexity***,
  indicates how fast an algorithm in question runs.

- ***Space efficiency***, also called ***space complexity***,
  refers to the amount of memory units required by the algorithm in addition to the space needed for its input and output.

In the early days of electronic computing,
both resources—time and space—were at a premium

What is algorithm optimality?

An algorithm can be said to be optimal if the function that describes its time complexity in the worst case is a lower bound of the function that describes the time complexity in the worst case of a problem that the algorithm in question solves.

What does optimal amount mean?

the best or most favorable point, degree, **amount**, etc., as of temperature, light, and moisture for the growth or reproduction of an organism. the greatest degree or best result obtained or obtainable under specific conditions.


Thank You