

**Data Structures &  
Algorithm Course Code:  
71203002002**

# Characteristics of Good Algorithms

What Makes an Algorithm Efficient, Practical, and Robust

- Correctness

- Efficiency

- Finiteness



- Definiteness

- Generality

- Portability

- Scalability

1



Unitedworld Institute of Technology **Why Do Characteristics Matter?**

- Multiple algorithms can solve the same problem •
- Choosing the best one improves time and space efficiency •
- Enhances maintainability and scalability
- Essential for real-world usability



Unitedworld Institute of Technology **Overview of Key Characteristics**

- **Correctness:** Produces correct output for all valid input
- **Efficiency:** Uses minimal time and memory
- **Finiteness:** Terminates after a finite number of steps

- **Definiteness:** Steps are unambiguous and clear
- **Generality:** Solves a class of problems
- **Portability:** Can be implemented across platforms
- **Scalability:** Performs well as input size grows

## Correctness

- **Definition:** Correct output for every possible valid input
- Test cases and formal verification techniques (loop invariants, proofs)

- Non-negotiable quality for any useful algorithm

## Efficiency

- **Time Efficiency:** Execution speed (e.g.,  $O(n)$ ,  $O(\log n)$ )
- **Space Efficiency:** Memory usage

- Better performance with growing data



Unitedworld Institute of Technology

## Finiteness

- **Definition:** Must terminate after finite steps
- Counterexamples include infinite loops (while True)

- Ensures the algorithm completes execution



## Definiteness

- **Definition:** Steps are precisely defined and unambiguous
- Prevents confusion during implementation

- Bad vs good examples clarify this importance



Unitedworld Institute of Technology

## Generality

- **Definition:** Solves a wide class of problems •



Not limited to one specific input or case •  
Enhances reusability of the algorithm



**Unitedworld Institute of Technology**

**Portability**

- **Definition:** Can be implemented in various languages •
- Works across different systems and environments •
- Improves flexibility and adoption

## Scalability

- **Definition:** Handles large input sizes efficiently •
- Performance degrades gracefully with size increase •
- Key for big data and large-scale systems

## Practical Application: Google Search

- **Correctness:** Fetches relevant web pages •
- Efficiency:** Handles billions of queries quickly •
- Scalability:** Maintains performance at internet scale •
- Portability:** Operates across global server clusters



**Unitedworld Institute of Technology**

## **Summary & Recap**

- **A good algorithm is:** Correct, Efficient, Finite, Clear, General, Portable, Scalable
- These traits make algorithms suitable for practical use
- Foundation for writing effective programs

