جامـــــعة المـــــستـقبـل
AL MUSTAQBAL UNIVERSITY

كلية العلـوم

قسـم علوم الذكاء الاصطناعي

المحاضرة الأولى

المادة: Searching and Sorting Algorithms
المرحلة: الثانية
اسم الاستاذ: م.م اية محمد حسين محمد علي

An introduction to Searching and Sorting Algorithms

Objectives

- Understand what Searching and Sorting Algorithms and why it is important.
- Explore the applications of Searching and Sorting Algorithms.
- Provide an overview of the course outline.

Characteristics of Searching and Sorting Algorithms

- Simple Concept

Easy to understand and based on clear logical steps.

- Improve Data Organization

Help organize data to make it easier to process and analyze.

- Efficiency-Oriented

Designed to reduce time and computational cost.

- Different Performance Levels

Algorithms vary in speed depending on data size and structure.

- Foundation of Computer Science

Considered core topics in algorithms and data structures.

- Scalable

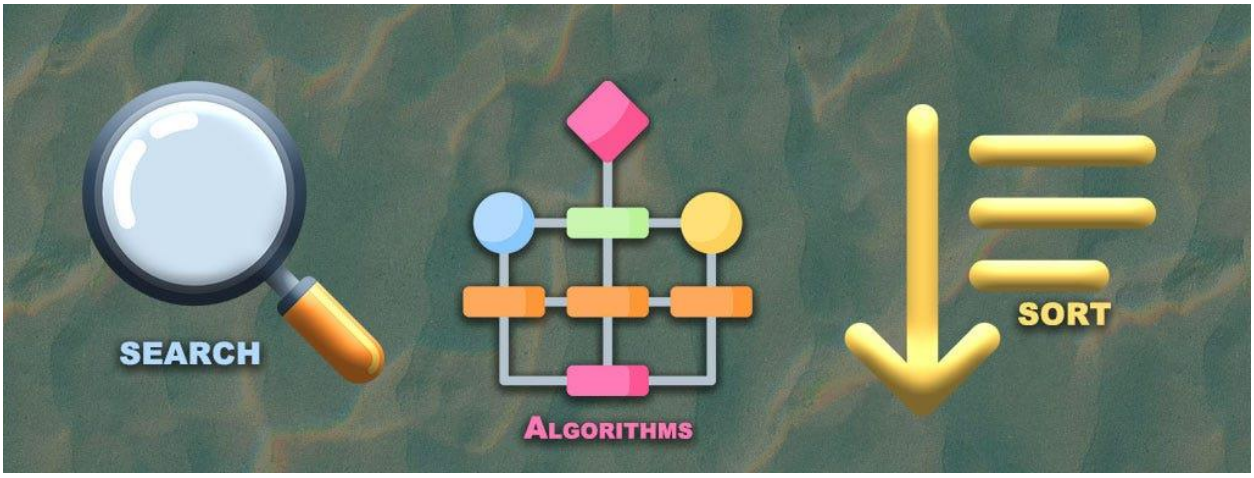
Work with small datasets and large datasets efficiently.

- Support Problem Solving

Help in solving real-world problems like searching, ranking, and ordering data.

- Widely Used in Applications

Used in databases, search engines, file systems, and software applications.



Real-World Applications

- Databases
- Search engines
- File management systems
- Data analysis
- Software applications

Recursion (Self-Calling)

Recursion is a programming technique where a function calls itself to solve a problem.
It is considered a mathematical and programming method that can be used instead of iteration (loops).

Many mathematical formulas can be implemented using recursion.

The factorial of a number is defined mathematically as:

Factorial of n ⟶ n! =n * (n−1) * (n−2)×(n−3) ⋯ *1

$$n! = \begin{cases} 1 & , n = 0 \\ n * (n-1)! & , n > 0 \end{cases}$$

Base Case:
When n = 0 → stop recursion and return 1

Recursive Case:
When n > 0 → call the function again with (n − 1)

***Recursion includes two main parts:**

- Base Case → stopping condition
- Recursive Case → function calls itself with smaller input

Examples:

1-How to Find the Power of X^m ?

$$X^m = \begin{cases} 1 & , m = 0 \\ x * x^{m-1} & , m > 0 \end{cases}$$

Code:

```
int power(int x, int m)
{
   if (m == 0)
      return 1;
   else
      return x * power(x, m - 1);
}
```

2.Fibonacci Sequence (Recursive Definition): In mathematics, the Fibonacci sequence, or Fibonacci numbers, is named after the Italian mathematician Leonardo Fibonacci. In this sequence, each number is equal to the sum of the two preceding numbers, except that the first number is 0 and the second number is 1.

The Fibonacci sequence is a series of numbers:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, …

Each number equals the sum of the two previous numbers.

$$F_n = F_{n-1} + F_{n-2}$$

$$F_1 = 1 \text{ و } F_0 = 0$$

Example: the 8th term is

the 7th term plus the 6th term:

F8 = F7 + F6

Code:

```
int fib(int n)
{
   if (n == 0) return 0;
   if (n == 1) return 1;

   return fib(n-1) + fib(n-2);
}
```

Resources

Author: MICHAEL McMillan. Title :" Data Structures and Algorithms Using C#" , 2007

Course Outline

1. Explain how recursion works in different types of function
2. what is tree and graph structure
3. Explain how to great a binary tree and different functions on binary tree
4. what is binary search tree and how to delete and insert nodes in a binary tree
5. Explain different algorithms for search and sorting algorithms performed on arrays