Algorithms And Data Structures

Kapilraj Trivedi

About me

- I have Master's degree in M.S.c Computer Science – Focus in Cyber Security
- I have 4 yrs of work experience in IT and Cyber Security
- I have worked with IBM, NII and ITK Engineering GmbH
- I have worked majorly in Agile, DevOps and CyberSec

Introduce Yourself

- Your Name
- Your city and country
- What is you're the thing that interests you more in computer science? Or your motivation for taking computer science? And Why?

Training outline

Module 1

INTRODUCTION

Module 2

DATA STRUCTURES

Module 3

COMPLEXITY ANALYSIS

Module 4

SORTING ALGORITHMS

Module 5

NON-LINEAR DATA STRUCTURES

Module 6

ABSTRACT DATA TYPES

Module 7

BINARY SEARCH

Module 8

DIJKSTRA'S ALGORITHM

Module 1: Introduction to Python

- Pre-requisite -
  - Python (Latest Version).
  - Visual Studio Code or any other python IDE

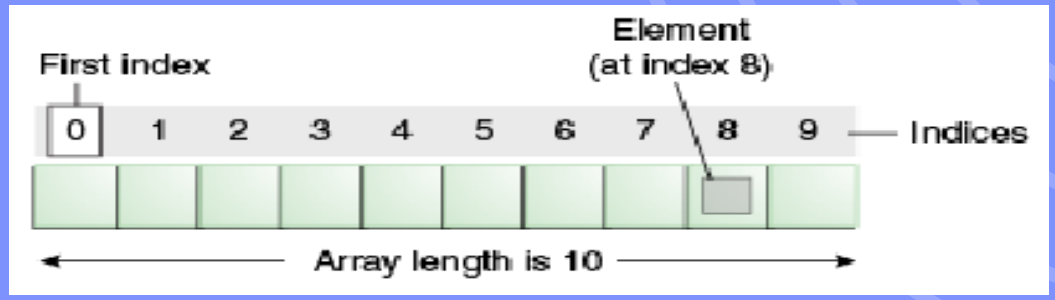
Module 1: Introduction to Data Structures

- In computer science, a data structure is a data organization, and storage format that is usually chosen for efficient access to data.
- Data Structures define how data is stored in RAM
- Many variations, each with advantages and disadvantages
- Strongly coupled to algorithmic complexity

Linear Data Structures

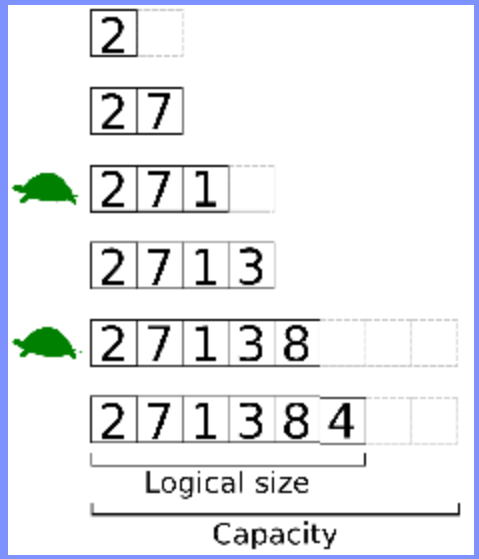
Arrays

- Linear, contiguous list of data
- Accessible by index
- Fixed-size
- N * d
- Supported by all major systems



Dynamic Arrays

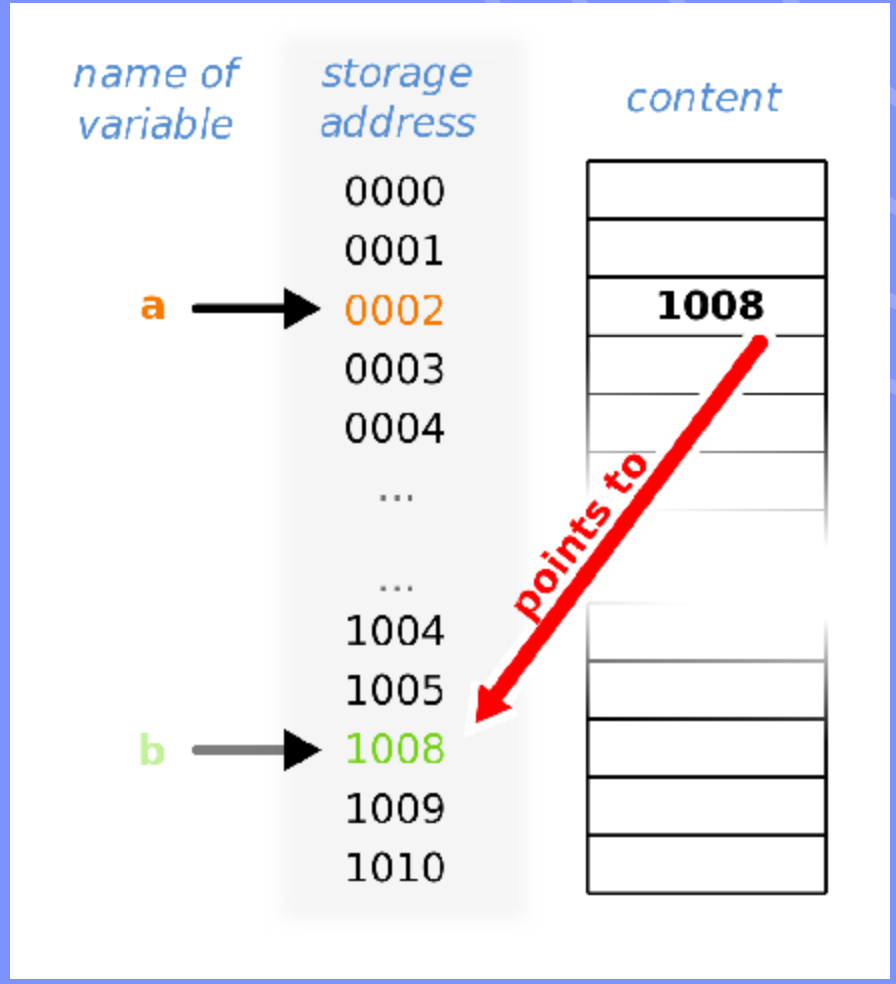
- Linear, contiguous list of data
- Accessible by index
- Resizable
- Python: List



Linear Data Structures

Pointers

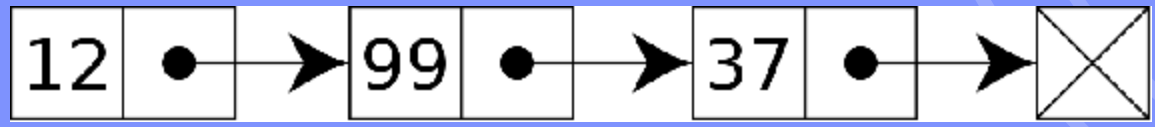
•Pointers are variables that store memory addresses rather than values directly.
•They are fundamental in low-level programming languages like C and C++ .
•They are not directly available in high-level languages like Python.
•In Python, pointers are often emulated using references or object references.



Linear Data Structures

Linked Lists

- Linear, contiguous list of data
- Accessible by iteration
- Resizable



Doubly Linked Lists

- Pointers both ways
- Uses more memory, but allows iteration both ways

Data Structures in python

Lists:

•Lists are ordered collections of items that can contain any data type. They are mutable, meaning you can change their elements after creation.

•**Scenarios**:

- ○Storing a collection of similar items, like a list of numbers or names.
- ○Maintaining a dynamic list of items where the size may change over time.

```
numbers = [1, 2, 3, 4, 5]
names = ["Alice", "Bob", "Charlie"]
mixed_list = [1, "apple", True, 3.14]
```

Tuples:

•Tuples are similar to lists but are immutable, meaning you cannot change their elements after creation.

•**Scenarios**:

- ○Storing a fixed collection of values, like coordinates or dimensions.
- ○Use when you don't want the data to be modified accidentally.

```
point = (10, 20)
dimensions = (100, 200, 300)
```

Data Structures in python

Dictionaries:

- Dictionaries are collections of key-value pairs. They are unordered, mutable, and can contain any data type.
- **Scenarios**:
  - Storing and retrieving data based on specific keys.
  - Representing structured data where each item has different properties.

```
person = {"name": "Alice", "age": 30, "city": "New York"}
employee = {"id": 1001, "name": "Bob", "department": "HR"}
```

Sets:

- Sets are unordered collections of unique elements. They are mutable like lists but can only contain immutable elements like strings or numbers.
- **Scenarios**:
  - Removing duplicates from a list of items.
  - Checking for membership or intersection between collections.

```
unique_numbers = {1, 2, 3, 4, 5}
vowels = {'a', 'e', 'i', 'o', 'u'}
```

Data Structures in python

Arrays (from NumPy):

•Arrays are homogeneous collections of elements, usually of the same data type. They are more efficient for numerical computations compared to lists

•**Scenarios**:

- ○Performing mathematical operations on large datasets efficiently.
- ○Working with multi-dimensional data like images or matrices.

```
import numpy as np
numbers_array = np.array([1, 2, 3, 4, 5])
matrix = np.array([[1, 2], [3, 4]])
```

Linked Lists:

•Sets are unordered collections of unique elements. They are mutable like lists but can only contain immutable elements like strings or numbers.

•**Scenarios**:

- ○Implementing algorithms like stacks, queues, or hash tables.
- ○Managing data where insertion, deletion, or traversal efficiency is crucial.

Let us Look at Data structures in python using for loops

Module 1: Introduction to Algorithms

- Algorithms are precise sequences of steps or instructions designed to solve specific problems or perform tasks.
- They are fundamental to computer science and are used in various applications, from simple calculations to complex data processing.
- **Importance of Algorithms**:
  - Algorithms form the backbone of computer programs and systems.
  - Understanding algorithms is crucial for developing efficient software solutions, optimizing processes, and solving problems effectively.Prototyping Model
- **Real-life Examples**: Algorithms are prevalent in everyday life. From following a recipe to navigate through traffic, we often rely on algorithms without even realizing it.

Thank you