DATA STRUCTURES

Introduction

Prof. Khalil Al-Shqeerat

CS Dept. – Qassim University

INTRODUCTION

- What is data?
  - Quantities, characters or symbols on which operations are performed by a computer
    - Stored on external memory (Hardware)
    - May be and transmitted in the form of electrical signals.
  - Example: C = A + B (A & B is data) C is information
    A & B are data
- When data becomes information?
  - If data is arranged in a systematic way then it gets a structure and become meaningful.
  - MEANINGFUL OR PROCESSED DATA IS CALLED **INFORMATION**.

🧠 أول شيء: ما هو الـ Data (البيانات)؟

تخيلي إن عندك أشياء بدون معنى لوحدها:

أرقام: 10 ، 20

حروف: A ، B

رموز

هذه كلها اسمها Data (بيانات)

👉 التعريف:

البيانات هي قيم أو رموز أو حروف الكمبيوتر يقدر يعالجها (يسوي عليها عمليات)

💡 مثال بسيط يخليك تفهم فورًا:

A = 10

B = 20

C = A + B

❗ (Data) بيانات → A و B

✅ (Information) معلومة → C = 30

ليش؟

لأنك استخدمت البيانات وسويت عليها عملية فصار لها معنى.

🧠 طيب متى تصير Data → Information؟

خلينا نقول عندك:

10 20 30

هذه مجرد أرقام مبعثرة = Data

لكن لو كتبنا:

درجات الطالب: 10 ، 20 ، 30

هنا صار لها معنى → صارت Information

🎯 القاعدة الذهبية (احفظها)

خام (بدون معنى) = Data

تنظيم أو معالجة = معنى + Information = Data

DATA

- Values or set of values.
- A data item refers to a single unit of values.
- Data items that are divided into sub items are group items; those that are not are called elementary items.
  - For example, an employee's name may be divided into three sub items – first name, middle name and last name
- An entity has specific attributes or properties that may be assigned values.
  - The values may be either numeric or non-numeric.

Data isn't meaningful unlike information, we have to perform operations for it to become meaningful (Info)

Sara Ahmed <- this is a data items

Sara <- first name (sub item)
Ahmed <- last names (sub item)

For example :

Entity: Student
Attributes or properties:
Student's Name (Data item type String)
Student's ID (Number or integer)
Student's Age (Number or integer)

Data = قيم
Data item = قيمة واحدة
Group item = ينقسم
Elementary item = ما ينقسم
Entity = شيء حقيقي (طالب مثلاً)
Attributes = صفاته
Values = قيم الصفات

WHAT IS A DATA STRUCTURE?

- A data structure is the systematic way to organize data so that it can be used efficiently.
- It is a group of data elements stored together under one name
  - Example: arrays
    - Instead of creating multiple variables of same type.
      - Storing strings is equivalent of storing a sequence of characters.
- It is also referred as "Abstract Data Type (ADT)"
  - ADT is a type (or class) for objects whose behavior is defined by a set of values and a set of operations
- The challenge is to select the most appropriate data structure for the problem

The goal of learning Data Structure:

طريقة منظمة لترتيب البيانات بحيث نستخدمها بكفاءة = Data Structure

💡 يعني مو بس نخزن بيانات...

لا، نخزنها بطريقة ذكية تساعدنا:

نبحث بسرعة

نضيف بسهولة

نحذف بدون مشاكل

هو تعريف يركز على:

وش نقدر نسوي (Operations)

مو كيف نسويه (Implementation)

🔥 مثال بسيط جدًا Stack : (المكدس)

نقول:

نضيف عنصر → push

نحذف عنصر → pop

لكن ما نهتم:

هل استخدمنا Array؟

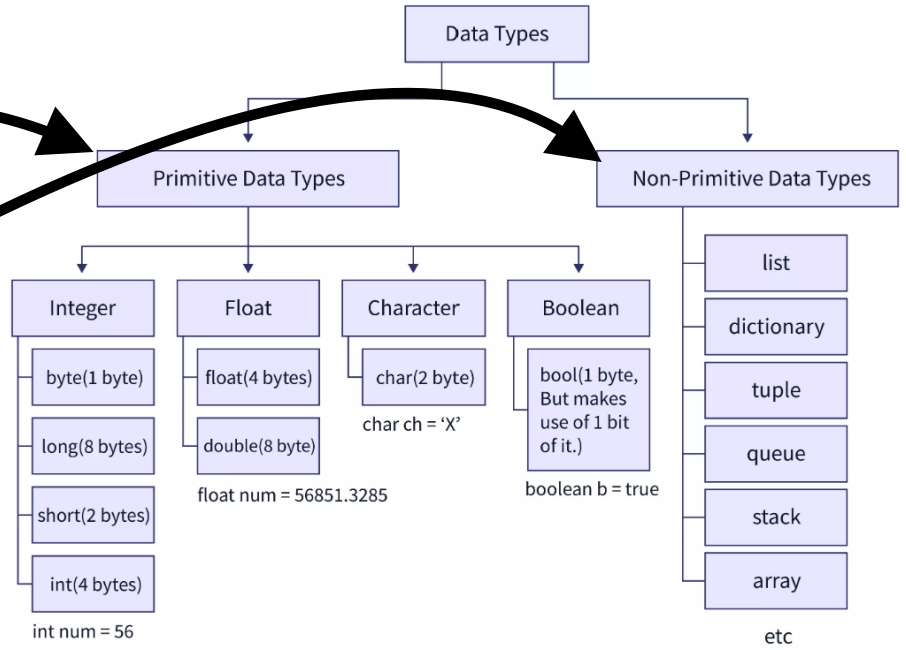
أو Linked List؟

CLASSIFICATION OF DATA STRUCTURES

- The data structure can be classified into two categories
  - primitive data structure
    - Storing the value of only one data type
  - Non-primitive data structure.
    - Storing the value of more than one data type

Primitive DS : Storing one value.
تخزن قيمة واحدة فقط

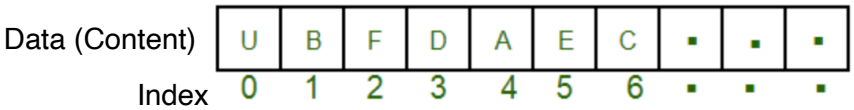
Non-Primitive DS : Storing multi values (or one).
تخزن قيمة واحدة او اكثر



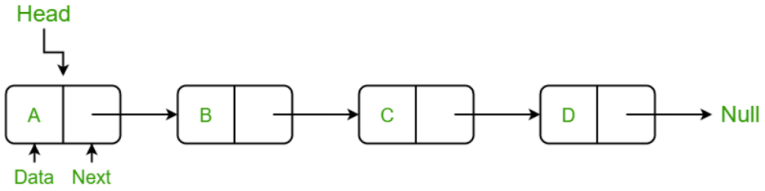
PHYSICAL DATA STRUCTURES hardware view

Arrays and Lists are the same (They both store Data of the same type), But they differ in how they store data.

- Actual implementations of data structures in memory.
  - Array, Linked list



Array



Linked list

Array Cons:
1- Faster Access to memory
2- Continuous (Data are in a sequence)

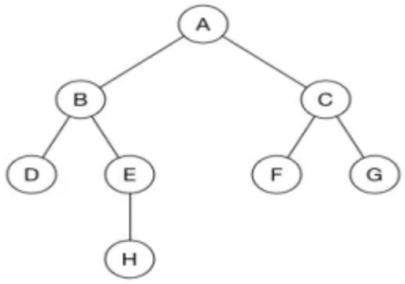
In Array we most determine the size of the array before.

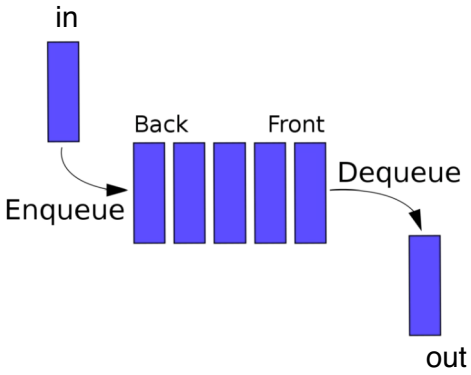
Linked list cons :
1- Dynamic size
2- Flexible
3- efficient insertion and deletion (Because of the dynamic size)

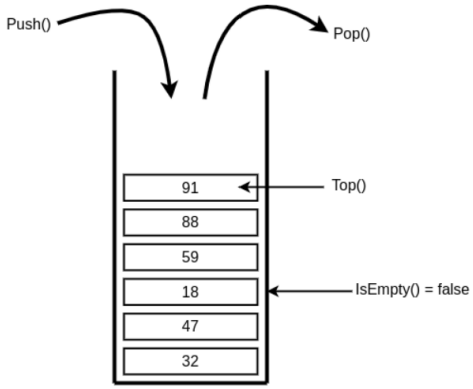
التعريف:
هي الطريقة الفعلية لتخزين البيانات في الذاكرة
يعني:
👉 كيف البيانات موجودة داخل الكمبيوتر
أمثلة:
🔷 Array
مخزن بشكل متصل (contiguous)
حجم ثابت
🔷 Linked List
مخزن بشكل غير متصل
حجم متغير

LOGICAL DATA STRUCTURES Programmer view

- Logical implementation of data over physical memory.
  - Stack, Queue, Tree







Tree

In hierarchy manner

Stack, Queue, Tree are a type of ADT

Queue

First in first out

التعريف:
هي طريقة التفكير أو تمثيل البيانات منطقيًا
يعني:
كيف نتخيل البيانات، مو كيف نخزنها
أمثلة:

- Stack

LIFO (اول من يدخل هو اخر من يطلع)

- Queue

FIFO (أول من يدخل هو اول من يطلع)

- Tree

شكل هرمي

Stack

Last in first out

Push: Insert element (Top -> Down)
Pop : delete (down -> top)

STATIC VS. DYNAMIC DATA STRUCTURES

- Static data structure
  - Collection of data in memory which have a fixed size
  - It can store a limited amount of elements or data in memory.
  - Array is an example of static data structure.
- Dynamic Data Structure
  - Collection of data in memory which do not have a fixed size
  - Its size can be modified during the operations performed on it
  - It can store a variable amount of elements or data in memory.
  - Linked List is an example of dynamic data structure.

The difference between them

التعريف:
هي بيانات حجمها ثابت (Fixed) وما يتغير
مثال: Array
💡 كيف تفهمها؟
تخيلي عندك صندوق فيه 5 أماكن فقط
❌ ما تقدر تزيد
❌ ما تقدر تنقص
المميزات:
بسيطة
سريعة في الوصول
العيوب:
حجم ثابت
ممكن يكون في هدر (Memory waste)

التعريف:
هي بيانات حجمها يتغير أثناء التشغيل
مثال: Linked List
💡 كيف تفهمها؟
تخيلي عندك سلسلة:
تقدر تضيف حلقة 🔗
أو تحذف حلقة
👉 مرنة جدًا
المميزات:
حجم متغير
مرونة عالية
العيوب:
أبطأ شوي
تستخدم ذاكرة أكثر (بسبب الـPointers)

LINEAR DATA STRUCTURES

- Data elements are arranged sequentially or linearly
- Every element is attached to its previous and next adjacent
- Single level is involved.
  - It can traverse all the elements in single run only.
- Easy to implement because computer memory is arranged in a linear way.
- Array, Stack, Queue, Linked list are examples of Linear data structures.

التعريف:
العناصر مرتبة واحد بعد الثاني (Sequence)
💡 تخيلها:
10 → 20 → 30 → 40
أمثلة:
Array
Stack
Queue
Linked List
مميزاتها:
سهلة
تمشي عليها خطوة خطوة

NON-LINEAR DATA STRUCTURES

- Data elements are not arranged sequentially or linearly.
- Hierarchical multiple levels are involved.
  - It can't traverse all the elements in single run only.
- Its implementation is complex.
- It utilizes computer memory efficiently in comparison to a linear data structure.
- Its examples are trees and graphs

عناصر مو على خط واحد، تكون على شكل شجرة أو شبكة

💡 تخيلها:

```
      10
    /   \
  20     30
    \
     40
```

أمثلة:

Tree 🌳

Graph

مميزاتها:

تمثل العلاقات المعقدة

تستخدم الذاكرة بشكل أفضل

DATA STRUCTURE OPERATIONS

هي العمليات اللي نسويها على البيانات
Data Structure داخل أي

- Different types of operations can be performed for the manipulation of data in every data structure.
  - Traversal: accessing each record/node exactly once so that certain items in the record may be processed.
  - Search: Finding the location of the desired node with a given key value, or to satisfy one or more conditions.
  - Insertion: Adding a new node/record to the structure.
  - Deletion: Removing a node/record from the structure.
  - Sort: arranging all data items in a particular order.
  - Merge: combining the data items of two different sorted list into a single sorted list.

Traversal → التنقل او المرور
Search → البحث
Insert → إضافة
Delete → حذف
Sort → ترتيب
Merge → دمج

HOW TO CHOOSE A DATA STRUCTURE?

- The choice of a appropriate data structure depends on the following factors:
  - Based on the operations that will be performed
    - A linked list used to perform insertions and deletions.
    - An array is used to perform indexing.
  - Based on the time complexity of the operations
    - For example, to perform searches frequently, a binary search tree is used.
  - Based on the space complexity of the operations
    - For example, to store a lot of data, you should use an array.
  - based on the amount of memory used
    - For example, to store a lot of data in memory, you should use a linked list

🔥 أهم فكرة في الجزء كله
ما في Data Structure أفضل دائمًا
الأفضل هو اللي يناسب المشكلة :
🔷 1. حسب العمليات
إدخال/حذف كثير : Linked List
وصول سريع : Array من خلال الـ Index

🔷 2. حسب السرعة (Time Complexity)
إذا تبي العمليات تكون سريعة
مثال:
بحث كثير → Binary Search Tree

🔷 3. حسب الذاكرة (Space)
أقل استهلاك → Array

أكثر استهلاك → Linked List
🔷 4. حسب توفر الذاكرة
ذاكرة متصلة → Array
ذاكرة مرنة → Linked List