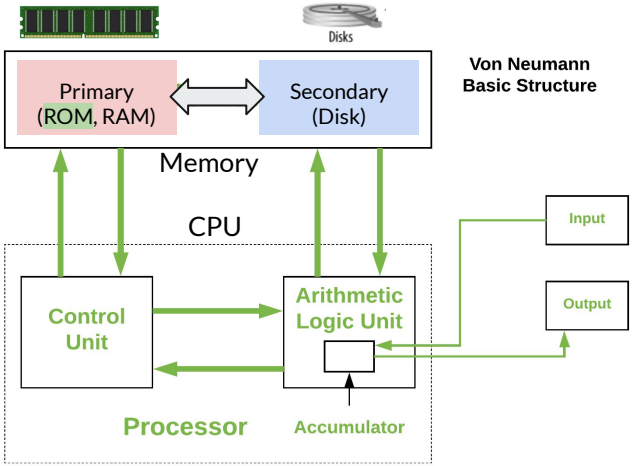
CIS 263 Introduction to Data Structures and Algorithms

Basic Data Structures: Array, Linked List, Stack, and Queue

Outline

- Brief introduction on **how a computer works** (architecture)
- **Computer memory**, the concept of our first data structure: **Array**

Computer Architecture basics

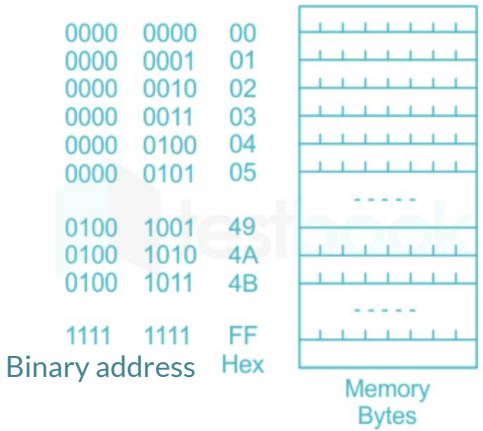


Von Neumann
Basic Structure

- In this course, our main focus will be mainly focused on the **memory management** part; more importantly the **primary memory**
- And **number of disk accesses** (latency delay)
- Through
  - Using appropriate data structures, and
  - Algorithms
- We are not doing any improvements over architecture etc.

https://www.geeksforgeeks.org/computer-organization-von-neumann-architecture/

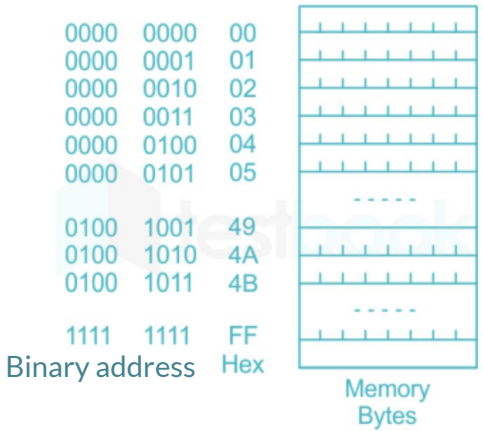
Memory, the concept of Array



- 8 bit memory (a simple case)

https://testbook.com/question-answer/a-memory-with-8-bit-data-bus-and-8-bit-address-bus--5e6e029ff60d5d4a3b0c5ce9

Memory, the concept of Array

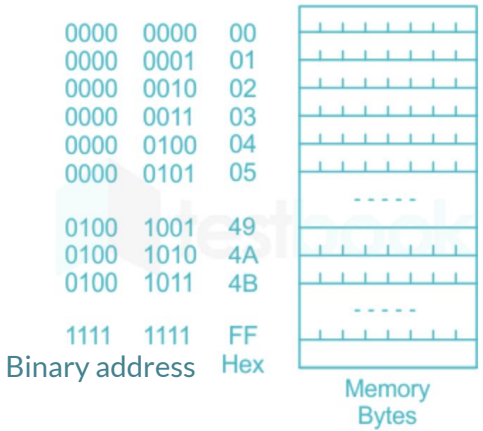


- 8 bit memory (a simple case)

Why 1 KB = 1024B not 1000B?

https://testbook.com/question-answer/a-memory-with-8-bit-data-bus-and-8-bit-address-bus--5e6e029ff60d5d4a3b0c5ce9

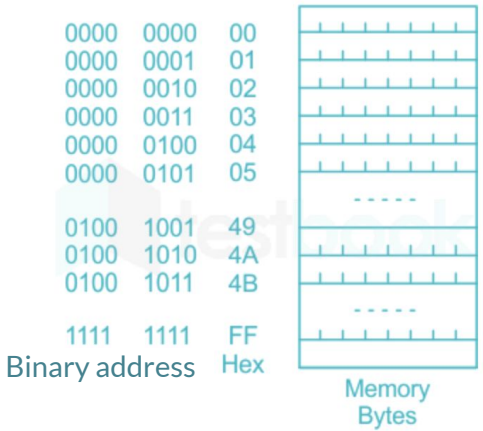
Memory, the concept of Array



- 8 bit memory (a simple case)
- **What's the total memory size?**

https://testbook.com/question-answer/a-memory-with-8-bit-data-bus-and-8-bit-address-bus--5e6e029ff60d5d4a3b0c5ce9

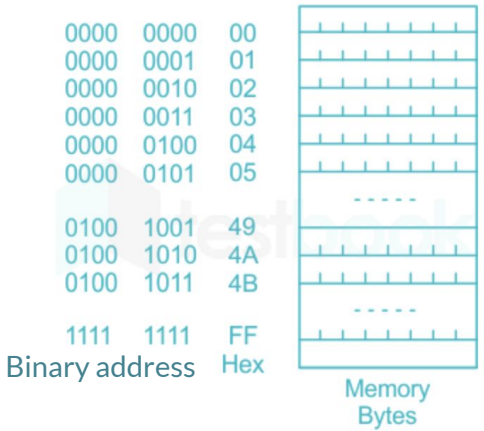
Memory, the concept of Array



- 8 bit memory (a simple case)
- **What's the total memory size?**
  - **2^8 or 1024 byte**

https://testbook.com/question-answer/a-memory-with-8-bit-data-bus-and-8-bit-address-bus--5e6e029ff60d5d4a3b0c5ce9

Memory, the concept of Array

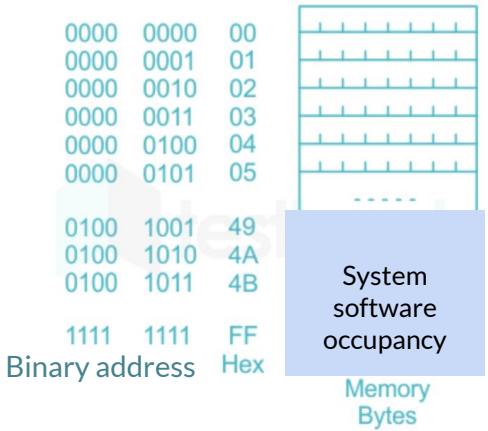


- 8 bit memory (a simple case)
- **What's the total memory size?**
  - **2^8 or 1024 byte**

Why 1 KB = 1024B not 1000B?

https://testbook.com/question-answer/a-memory-with-8-bit-data-bus-and-8-bit-address-bus--5e6e029ff60d5d4a3b0c5ce9

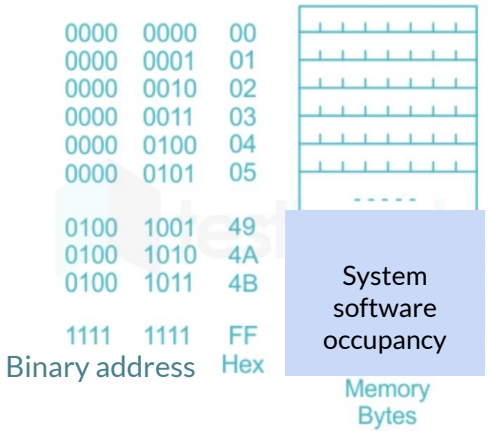
Memory, the concept of Array



- 8 bit memory (a simple case)
- What's the total memory size?
  - 2^8 or 1024 byte
- **Addressing is inherent (consecutive, a sequence)**

https://testbook.com/question-answer/a-memory-with-8-bit-data-bus-and-8-bit-address-bus--5e6e029ff60d5d4a3b0c5ce9

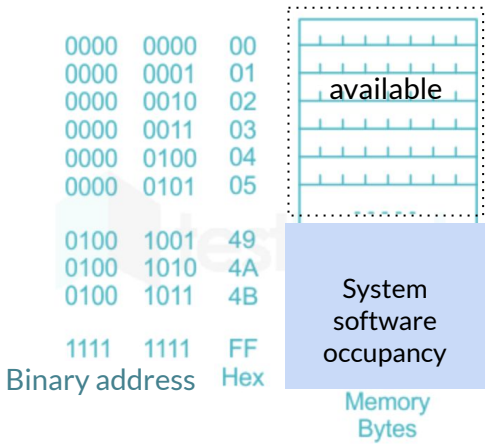
Memory, the concept of Array



- 8 bit memory (a simple case)
- What's the total memory size?
  - 2^8 or 1024 byte
- **Addressing is inherent (consecutive, a sequence)**
- **Let's assume our system softwares reserved part of the memory like**

https://testbook.com/question-answer/a-memory-with-8-bit-data-bus-and-8-bit-address-bus--5e6e029ff60d5d4a3b0c5ce9

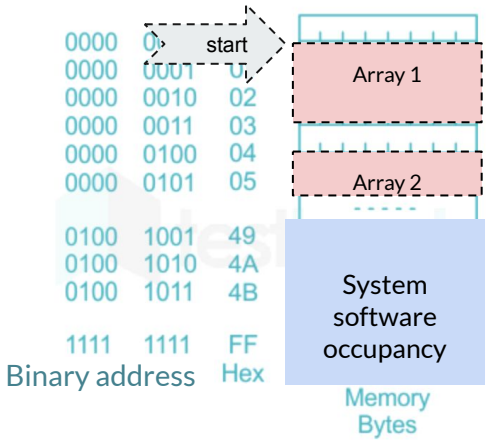
Memory, the concept of Array



- Addressing is inherent (consecutive, a sequence)
- Let's assume our system softwares reserved part of the memory like

https://testbook.com/question-answer/a-memory-with-8-bit-data-bus-and-8-bit-address-bus--5e6e029ff60d5d4a3b0c5ce9

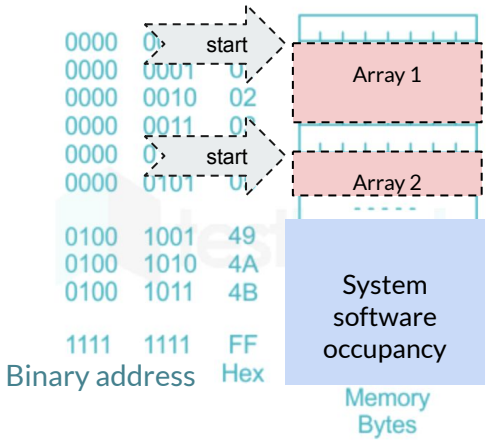
Memory, the concept of Array



- Addressing is inherent (consecutive, a sequence)
- Let's assume our system softwares reserved part of the memory like
- **Array** (our first data structure) meaning holding a fixed block whether you use it or not.

Image courtesy

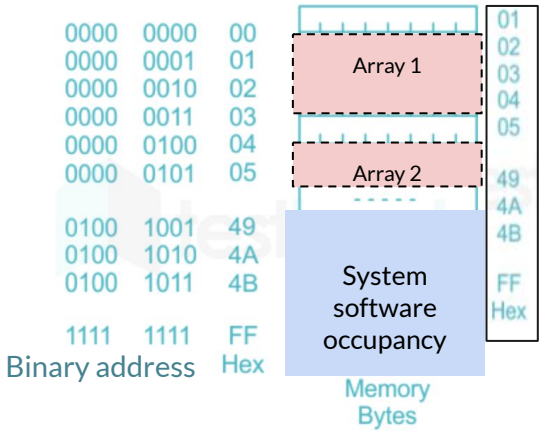
Memory, the concept of Array



- Addressing is inherent (consecutive, a sequence)
- Let's assume our system softwares reserved part of the memory like
- **Array** (our first data structure) meaning holding a fixed block whether you use it or not.

Image courtesy

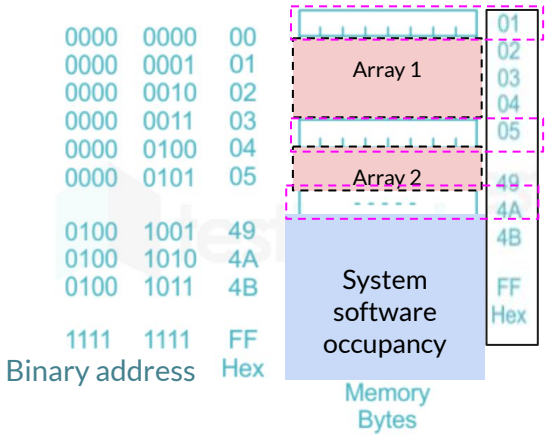
Memory, the concept of Array



- Addressing is inherent (consecutive, a sequence)
- **Fragmented memory**

Image courtesy

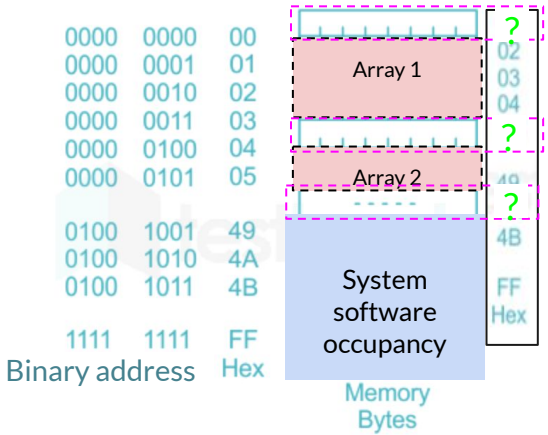
Memory, the concept of Array



- Addressing is inherent (consecutive, a sequence)
- Fragmented memory
- **Garbage collection**

Image courtesy

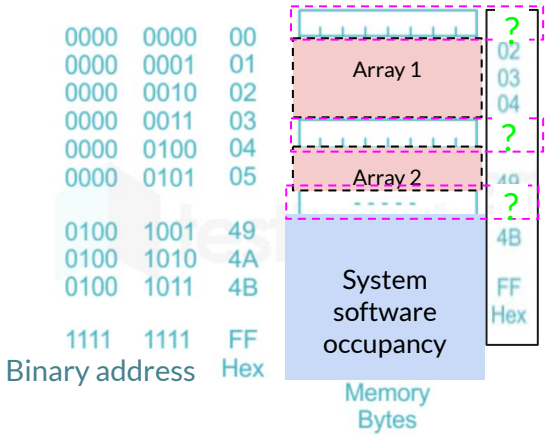
Memory, the concept of Array



- Addressing is inherent (consecutive, a sequence)
- Fragmented memory
- Garbage collection
- **Let's try, another data structure, linked list**

Image courtesy

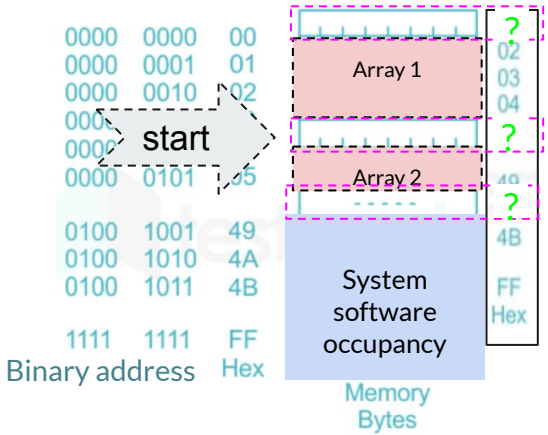
Memory, the concept of Array



- Addressing is inherent (consecutive, a sequence)
- Fragmented memory
- Garbage collection
- **Let's try, another data structure, linked list**
  - **You can start from anywhere**

Image courtesy

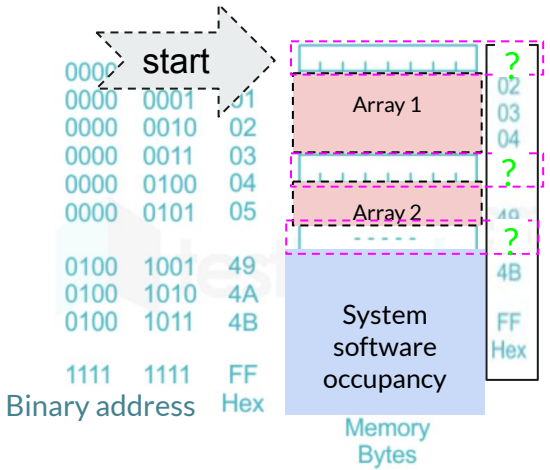
Memory, the concept of Array



- Addressing is inherent (consecutive, a sequence)
- Fragmented memory
- Garbage collection
- **Let's try, another data structure, linked list**
  - **You can start from anywhere**

Image courtesy

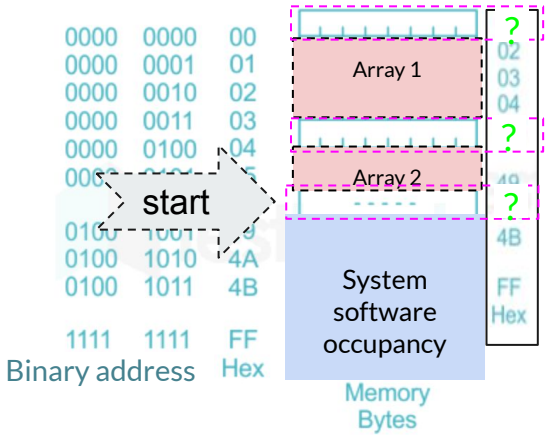
Memory, the concept of Array



- Addressing is inherent (consecutive, a sequence)
- Fragmented memory
- Garbage collection
- **Let's try, another data structure, linked list**
  - **You can start from anywhere**

Image courtesy

Memory, the concept of Array

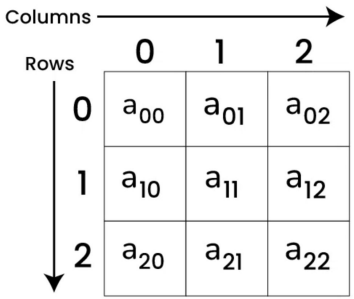


- Addressing is inherent (consecutive, a sequence)
- Fragmented memory
- Garbage collection
- **Let's try, another data structure, linked list**
  - **You can start from anywhere**

Image courtesy

Array

(2D) Array



Img src

- **Array** (our first data structure) meaning holding a fixed block whether you use it or not.
- Addressing is inherent (consecutive, a sequence)
- Operations:
  - Accessing an item
  - Query (if an item exists)
  - Insertion
  - Deletion
- Accessing an item is time independent; why?

Array

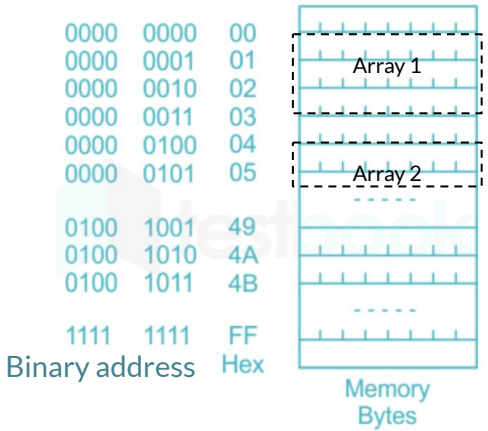
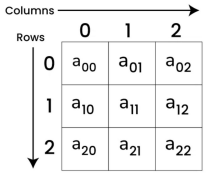
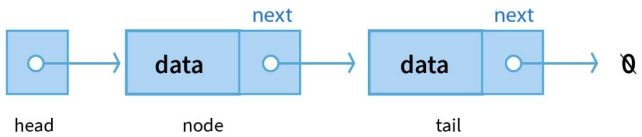


Image courtesy



- **Array** (our first data structure) meaning holding a fixed block whether you use it or not.
- Addressing is inherent (consecutive, a sequence)
- Operations:
  - Accessing an item
  - Query (if an item exists)
  - Insertion
  - Deletion
- Accessing an item is time independent; why?

Linked List



Img src

- Fragmented block as available
- Addressing is implicit within the data structure itself
- Operations:
  - Accessing an item
  - Query (if an item exists)
  - Insertion
  - Deletion
- Accessing an item is time dependent

Stack

Fill in the blank

A kid is reading on a _ _____ of books.

Stack

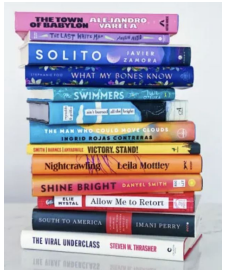
Fill in the blank

A kid is reading on a _pile_ of books.

Stack

Fill in the blank.

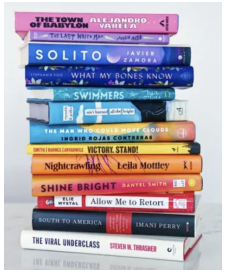
A kid is reading on a _pile__ of books.



I found a _____ of books.

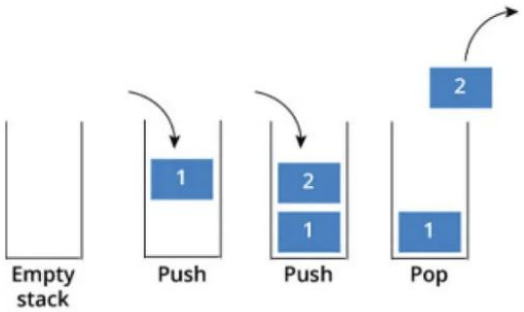
Stack

A kid is reading on a _pile__ of books.



I found a __stack___ of books.

Stack

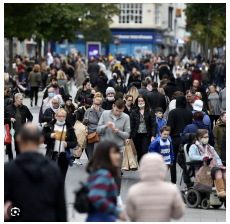


Img src

- An abstract Data Type (you can use an Array or LL to define a queue)
- Operations:
  - Accessing an item
  - Query (if an item exists)
  - Insertion (**push**)
  - Deletion (**pop**)

Queue

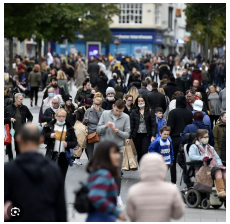
Fill in the blank



A _____ of people.

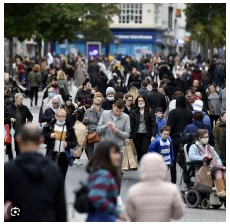
Queue

Fill in the blank



A _group__ of people.

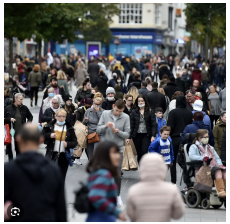
Queue



A _group__ of people.

A ____ of people.

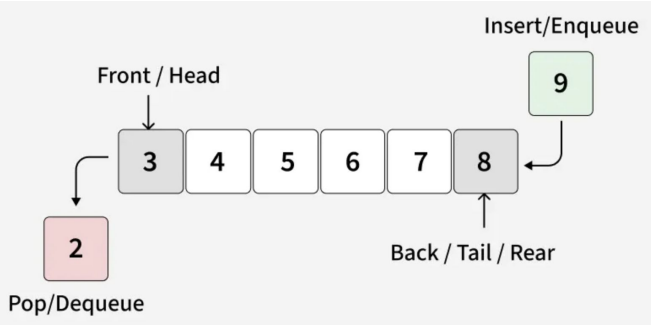
Queue



A _group__ of people.

A __queue___ of people.

Queue



Img src

- An abstract Data Type (you can use an Array or LL to define a queue)
- Operations:
  - Accessing an item
  - Query (if an item exists)
  - Insertion
  - Deletion