Data Structures

Logical and physical View

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Recall: 2D Arrays in memory

- What happens when u create such 2D array?
- In row-major order style, the **physical memory** will be first data row, then 2nd and so on.
- In col-major order, data is ordered column by column
- Observe: the array eventually just *consecutive numbers* in memory

```
int num[3][4] = {
  {1, 2,  3,  4},
  {5, 6,  7,  8},
  {9, 10, 11, 12}
};
```

	<—— row 0 ——>				<—— row 1 ——>				<—— row 2 ——>			
value	1	2	3	4	5	6	7	8	9	10	11	12
address	1000	1002	1004	1006	1008	1010	1012	1014	1016	1018	1020	1022

Img Src

Array: A Physical Data Structure

- In C++, **array** is a built-in physical data structure:
  - **Physical**: data stored directly in the **memory**
  - Data structure: **organized** data + **operations** over them
- Vector Data Structure
  - Internally, is just a dynamic array, so another physical data structure
- Static array is created on the **stack**
  But dynamic array is on the **heap**

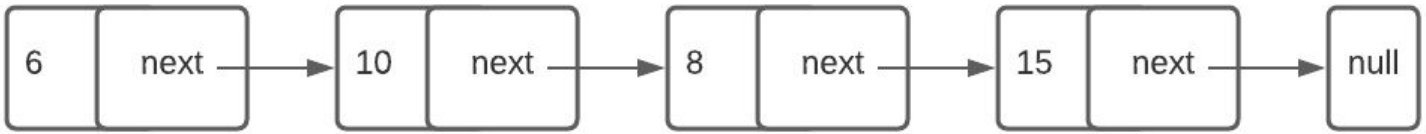
```
int numbers[] {1, 6, 10, 5};
numbers[0] = -3;
cout<<numbers[2]<<"\n";

float arr[7][10] {};
```

2D array: logical and physical view

- What is 2D array? Just a **table** where we access using [][]
  - This is called a **logical** view
  - **No care** how is that exactly in low-level memory
- Physical: How is it organized in memory
  - Eventually, all N*M data are consecutive in memory
  - But data can be either **row-major** order or **column-major** order
    - DIfferent *math equations* to locate arr[i][j]
- 1 logical view, but 2 physical views

Linked List

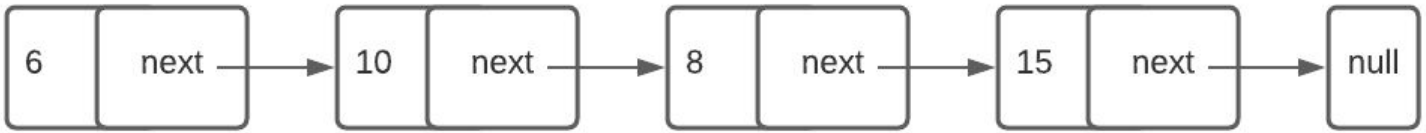
- Linked lists represents the memory directly through its nodes that can grow or shrink
- In other words, it **defines** how the **memory** should be organized for **storing** the data
  - We provide basic utilities to: insert (front/end/key) element or delete it
- Linked list is another Physical data structure
- Array is **contiguous** memory but linked list is not.



The memory

- The are ONLY 2 ways to utilize memory
  - A consecutive block of data through an array
  - Scattered elements, such as separate variables that we may link as in linked list
- Array & Linked List are the 2 main physical data structures

1	2	3	4	5	6	7	8	9	10	11	12
1000	1002	1004	1006	1008	1010	1012	1014	1016	1018	1020	1022



Logical Data Structures

- In practice, we need several data arrangements and operations over them
  - Think: **Restaurant queue**: We need the first in to be the first served (FIFO)
- We think ADT wise. We need a data structure that supports X, Y, Z, etc
  - What first. How later.
- These kind of data structures are typically logical ones
  - Later they will be implemented based on physical data structures, e.g. **array or linked nodes**
  - Big picture view = logical (what). The actual low-level memory arrangements = physical (how).
- Most of the data structures you will create/use are logical
  - Stack, Queue, Heaps, Trees, Hash Tables
- If confused, that is ok. You will get the difference over time

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."