



មហាវិទ្យាល័យវិស្វកម្ម  
FACULTY OF ENGINEERING

# Data Structure & Algorithm

## Lecture 5

### Order Array: The Insertion Sort

Chhoeum Vantha, Ph.D.

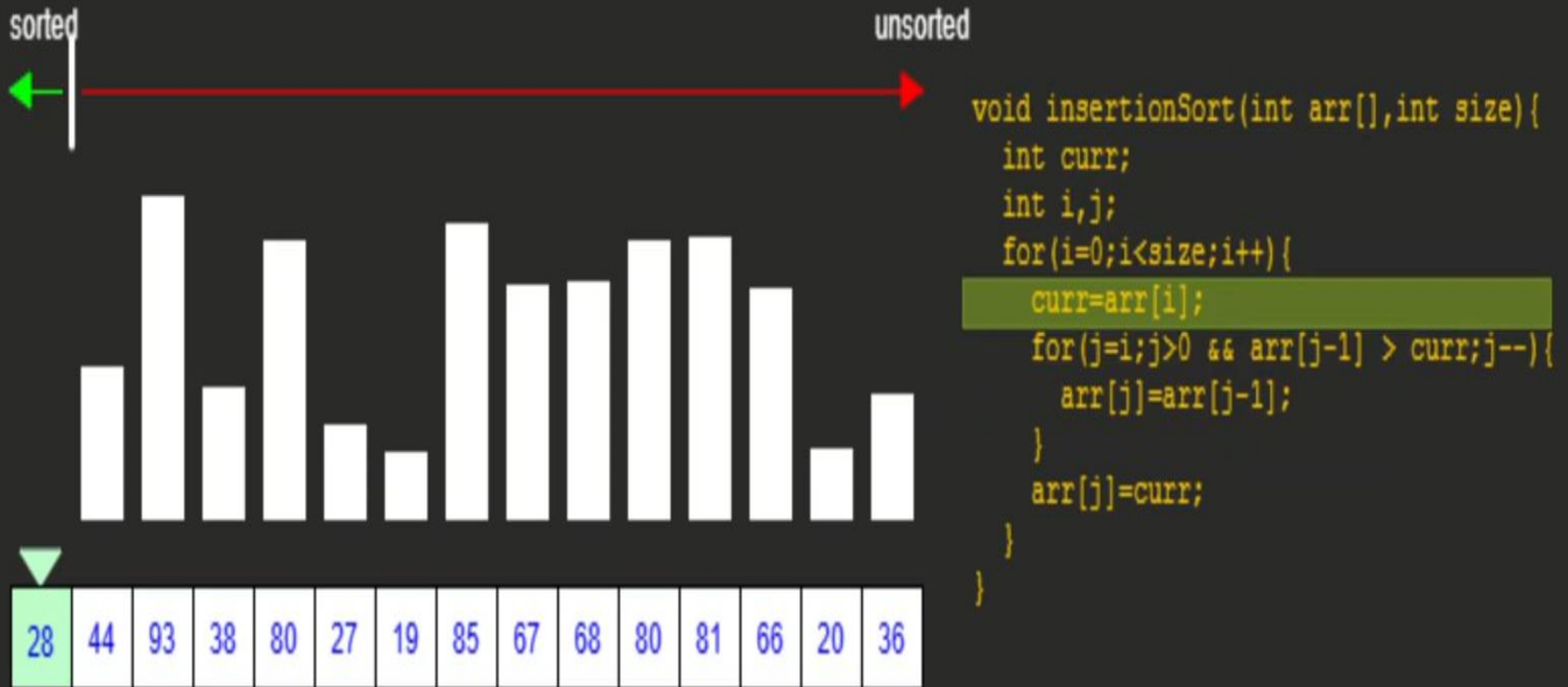
Telecom & Electronic Engineering

# Content

- Unordered Array
- Ordered array
  - Bubble Sort
  - Insertion Sort
  - Selection Sort
  - Quick Sort

# Insertion Sort

# Describe the video: What?



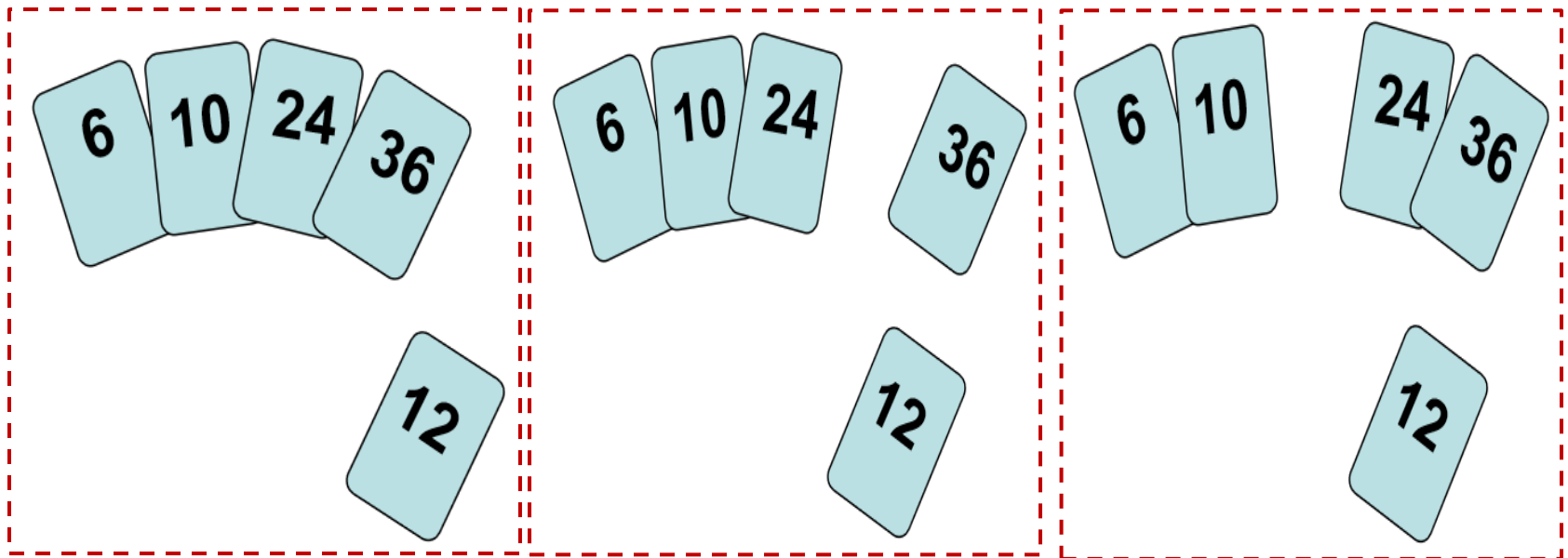
# Insertion Sort

Idea: like sorting a hand of playing cards

- Start with an empty left hand and the cards facing down on the table.
- Remove one card at a time from the table, and insert it into the correct position in the left hand
  - compare it with each of the cards already in the hand, from right to left
- The cards held in the left hand are sorted
  - these cards were originally the top cards of the pile on the table

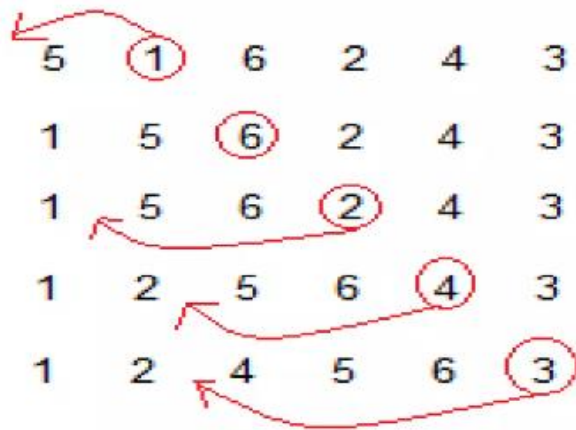
# Insertion Sort

- To **insert 12**, we need to make room for it by moving first 36 and then 24.



# Insertion Sort

5	1	6	2	4	3
---	---	---	---	---	---



( Always we start with the second element as key.)

Lets take this Array.

As we can see here, in insertion sort, we pick up a key, and compares it with elemnts ahead of it, and puts the key in the right place

5 has nothing before it.

1 is compared to 5 and is inserted before 5.

6 is greater than 5 and 1.

2 is smaller than 6 and 5, but greater than 1, so its is inserted after 1.

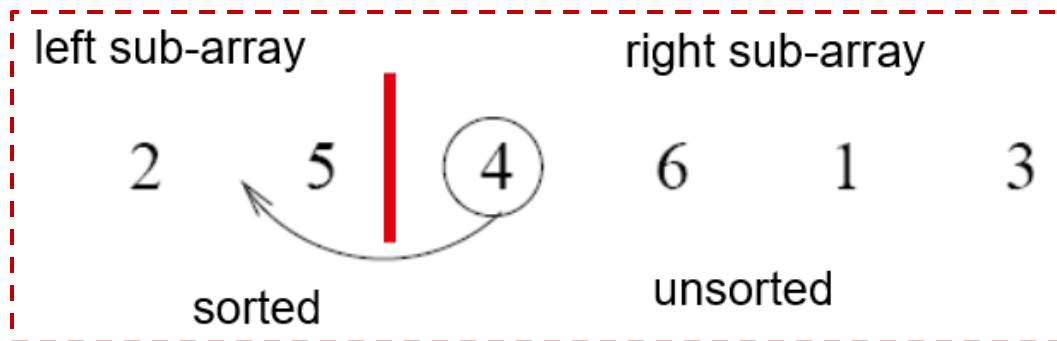
And this goes on...

# Insertion Sort

- input array



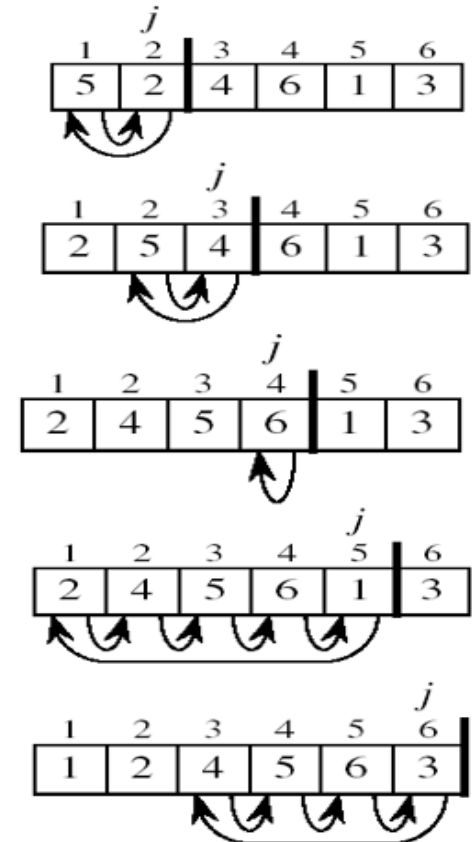
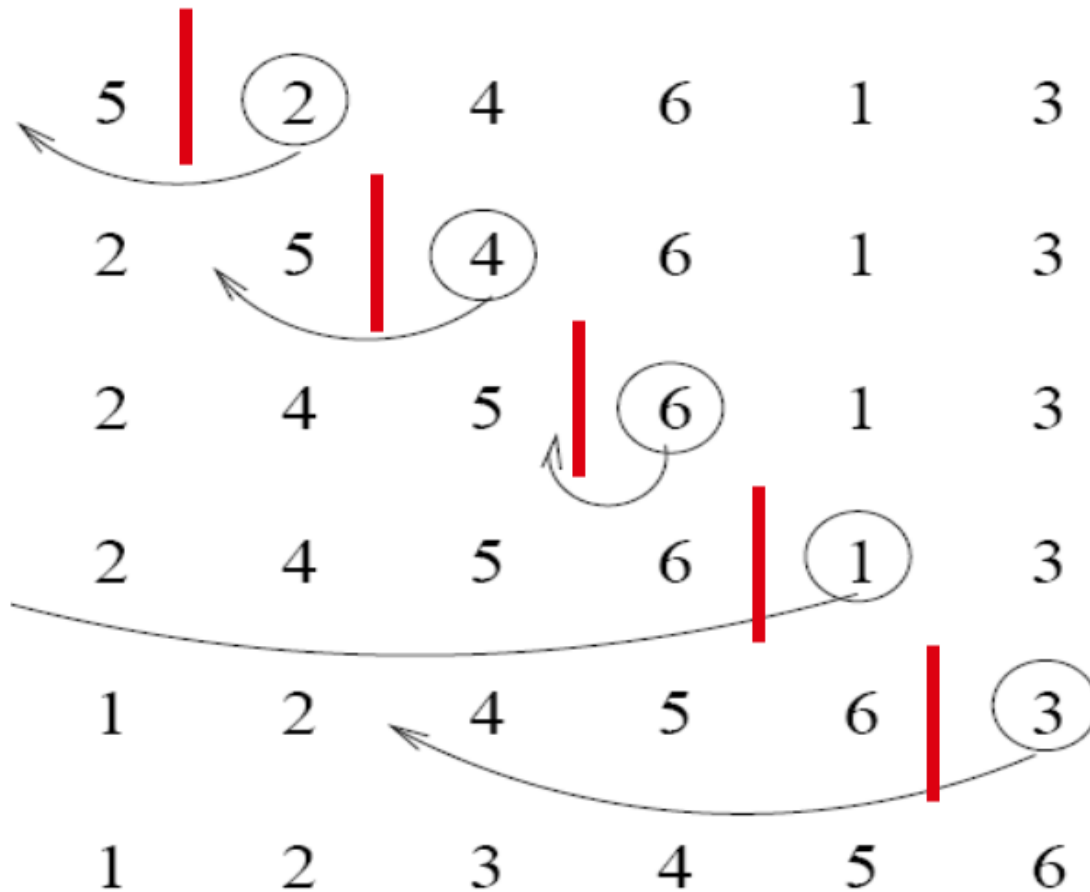
- at each iteration, the array is divided in two sub-arrays:





# Insertion Sort

- How you do it in C++?



# Insertion Sort

- It is a **simple Sorting algorithm** that sorts the array by **shifting elements**
- The array is virtually **split** into a **sorted** and an **unsorted part**.
- **Values** from the **unsorted** part are picked and **placed** in the correct position in the **sorted part**.

# Insertion Sort Algorithm in C++

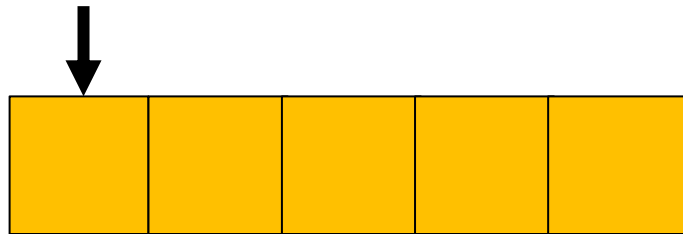
- How you do it in C++?

# Vectors

- Vectors are the same as **dynamic arrays** with the **ability to resize themselves** automatically when an element is **inserted or deleted**, with their **storage being handled** automatically by the **container**.
- Vector elements are **placed in contiguous storage** so that they can be accessed and traversed using iterators.

# Syntax of Vectors in C++

element




vector <int> v(5);

Type

Name

size

vector<int> v1 {5,2}; → 

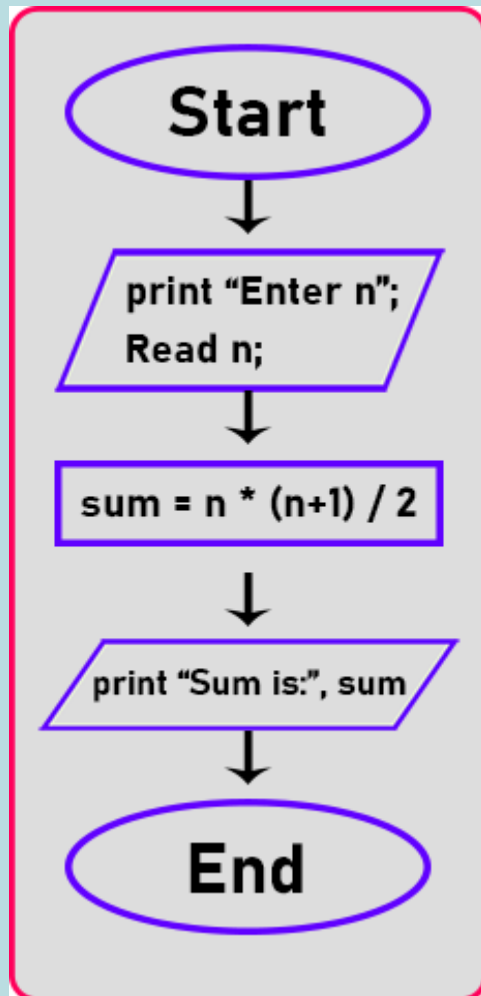
vector<int> v2 (5,2); → 

number of elements

default element value

# W5 – Lab 5

## Example: Flow chart to ADD to numbers



```
#include <iostream>
using namespace std;
int main()
{
    int n, sum;
    cout << "Enter n: ";
    cin >> n;
    sum = n * (n + 1) / 2;
    cout << "Sum is: " << sum;
    return 0;
}
```

# Exercise

1. Create a class of array to store data aof ny type, you want (such as: int, char, float,...)
2. Write a function, which will sort data of the array by ascending or descending with the Insertion sort algorithm (Option ascending or descending will be input by the user);
3. Explain codes by comments
4. Draw a flow chart of your source codes.



Thanks!