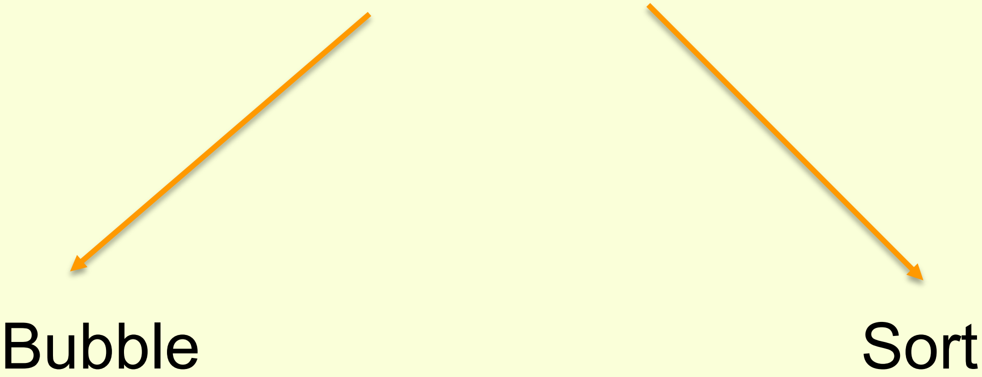
Presented By : Himanshu Dangwal
NetID : rm1716
Email : hdangwal@horizon.csueastbay.edu

Bubble Sort

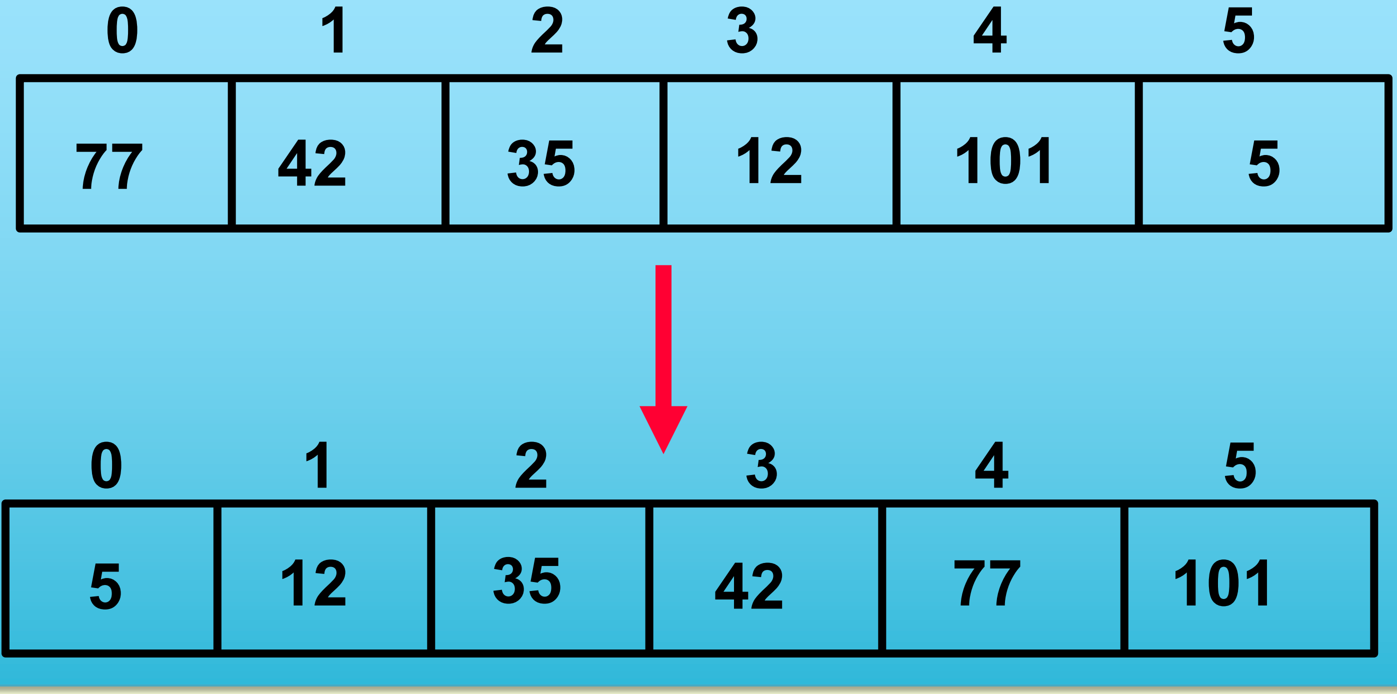


Bubble

Sort

Sorting

Sorting takes a list of elements and makes it an ordered one.



0	1	2	3	4	5
77	42	35	12	101	5

0	1	2	3	4	5
5	12	35	42	77	101

How does it sort the list of elements ?

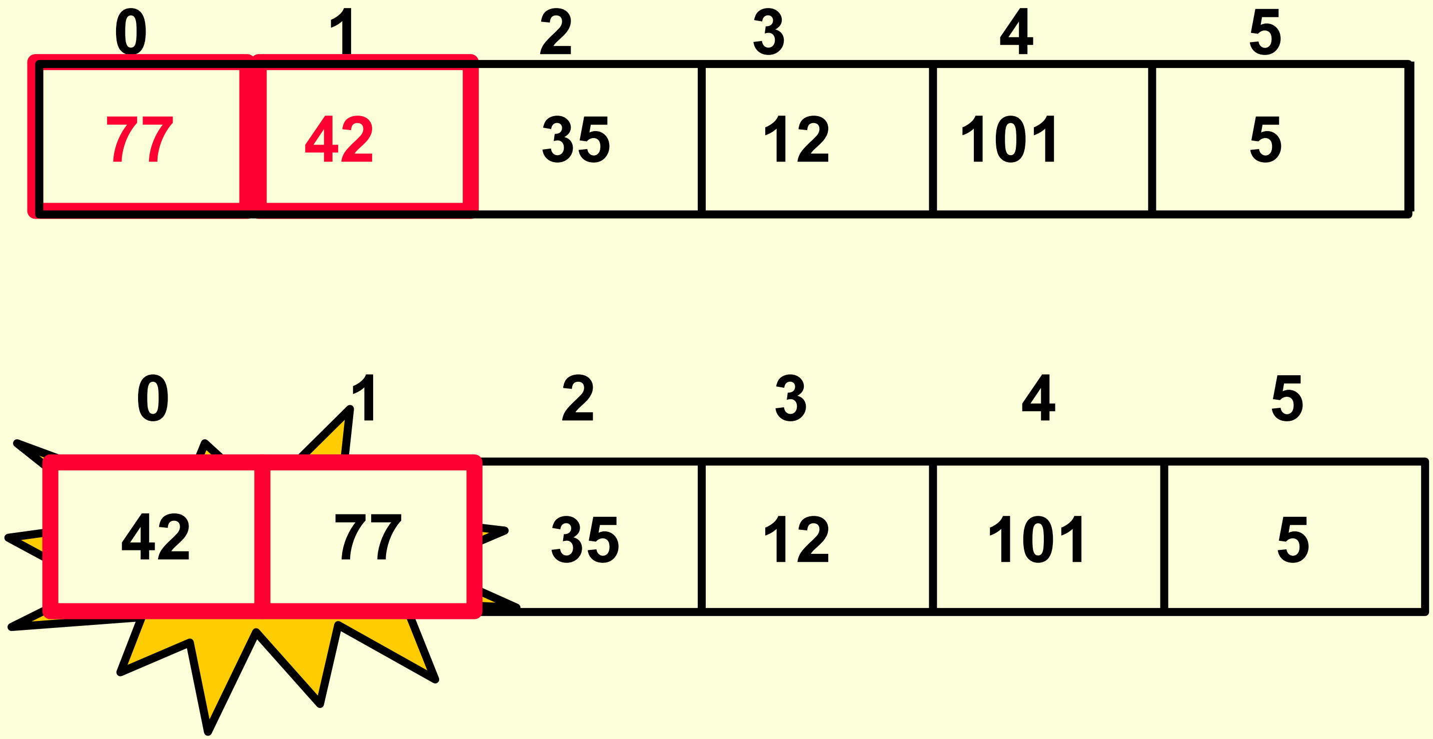
"Bubbling Up" the Largest Element

In every step it bubble up the largest element of the unordered list towards the end

0	1	2	3	4	5
77	42	35	12	101	5

"Bubbling Up" the Largest Element

- How does it bubbles up ?
    - It compares adjacent values, and it they are not in order, it swaps them.



0	1	2	3	4	5
77	42	35	12	101	5

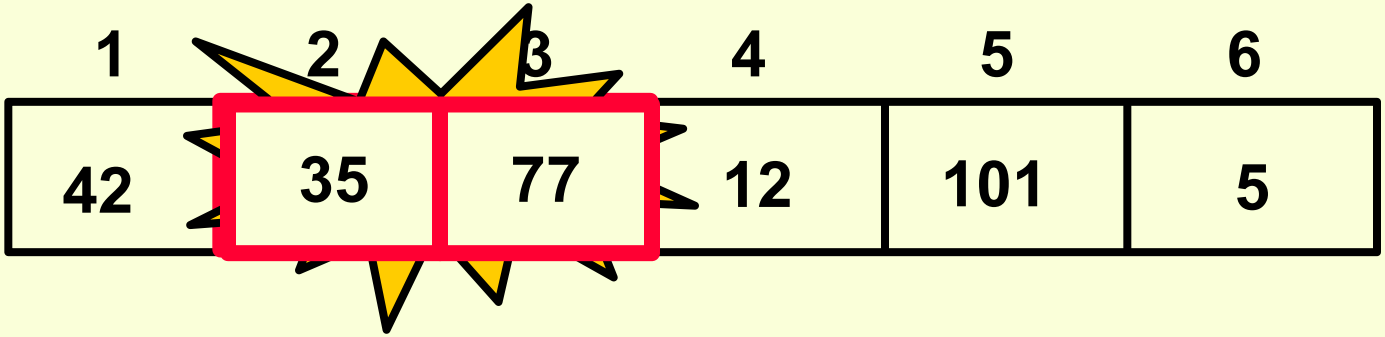
0	1	2	3	4	5
42	77	35	12	101	5

Let’s visualize this

https://learn.newtonschool.co/visuals/bubble_sort/0

"Bubbling Up" the Largest Element

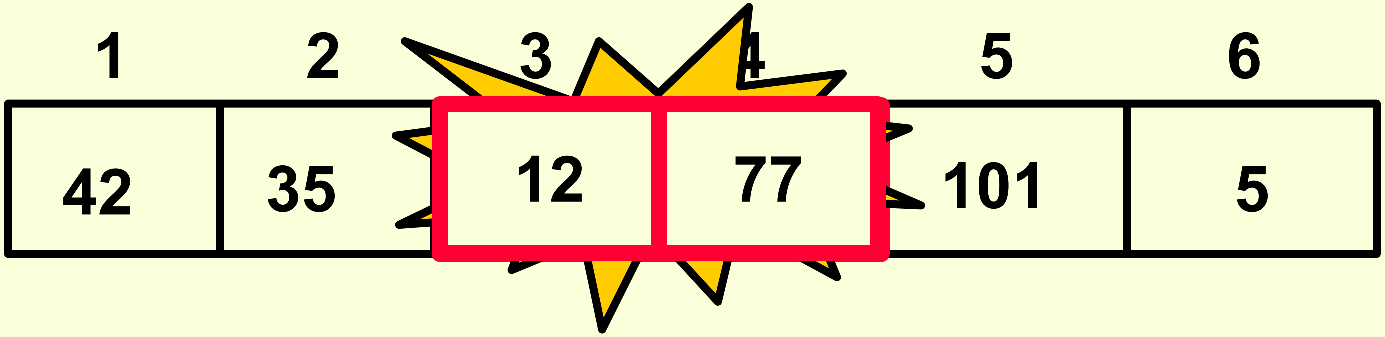
- It compares adjacent values, and it they are not in order, it swaps them.



1	2	3	4	5	6
42	35	77	12	101	5

"Bubbling Up" the Largest Element

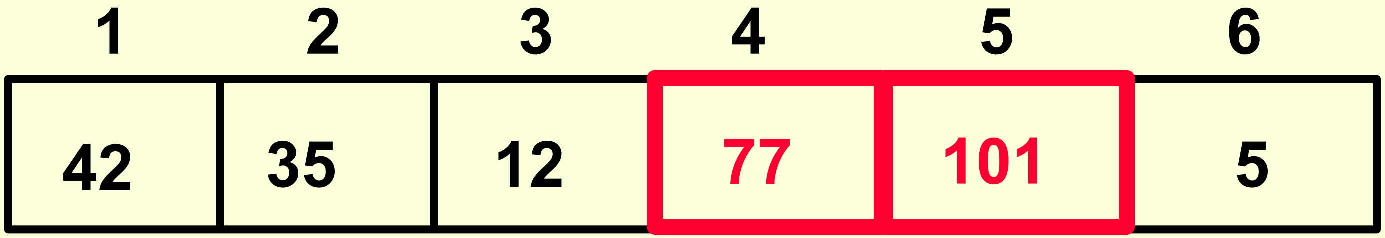
- Traverse a collection of elements
  - Move from the front to the end
  - "Bubble" the largest value to the end using pair-wise comparisons and swapping



1	2	3	4	5	6
42	35	12	77	101	5

"Bubbling Up" the Largest Element

- Traverse a collection of elements
  - Move from the front to the end
  - "Bubble" the largest value to the end using pair-wise comparisons and swapping

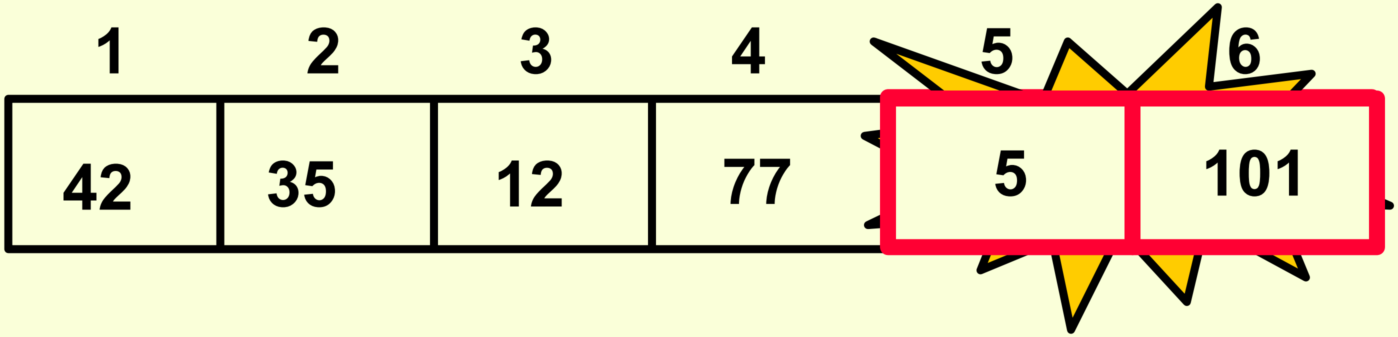


1	2	3	4	5	6
42	35	12	77	101	5

No need to swap

"Bubbling Up" the Largest Element

- Traverse a collection of elements
  - Move from the front to the end
  - "Bubble" the largest value to the end using pair-wise comparisons and swapping



"Bubbling Up" the Largest Element

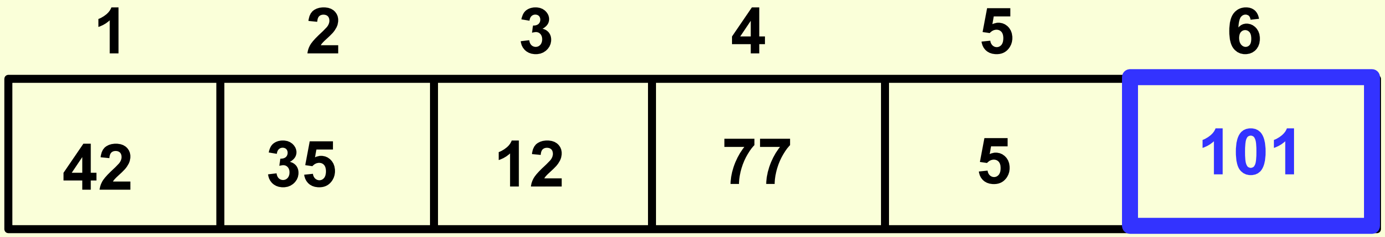
- After 1st iteration over the unsorted list, the largest element gets bubbled up towards the end.

0	1	2	3	4	5
42	35	12	77	5	101

Largest value correctly placed

Items of Interest

- Notice that only the largest value is correctly placed
- All other values are still out of order
- So we need to repeat this process



1	2	3	4	5	6
42	35	12	77	5	101

Largest value correctly placed

Repeat “Bubble Up” How Many Times?

- **If we have N elements…**
- **And if each time we bubble an element, we place it in its correct location…**
- **Then we** repeat the “bubble up” process N – 1 times.
- **This** guarantees we’ll correctly place all N elements.

“Bubbling” All the Elements

0	1	2	3	4	5
42	35	12	77	5	101

0	1	2	3	4	5
35	12	42	5	77	101

0	1	2	3	4	5
12	35	5	42	77	101

0	1	2	3	4	5
5	12	35	42	77	101

N - 1

Reducing the Number of Comparisons

1	2	3	4	5	6
77	42	35	12	101	5

1	2	3	4	5	6
42	35	12	77	5	101

1	2	3	4	5	6
35	12	42	5	77	101

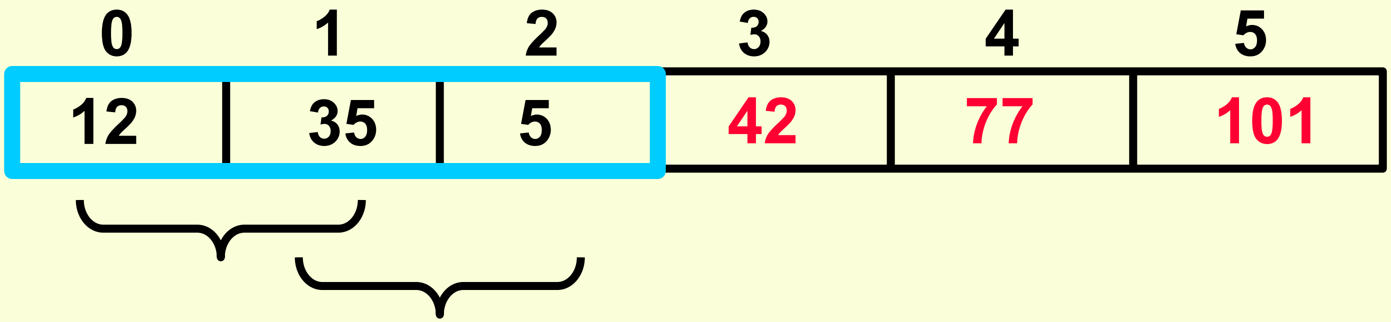
1	2	3	4	5	6
12	35	5	42	77	101

1	2	3	4	5	6
12	5	35	42	77	101

Reducing the Number of Comparisons

- For every new bubble up the number of comparisons are getting reduced.

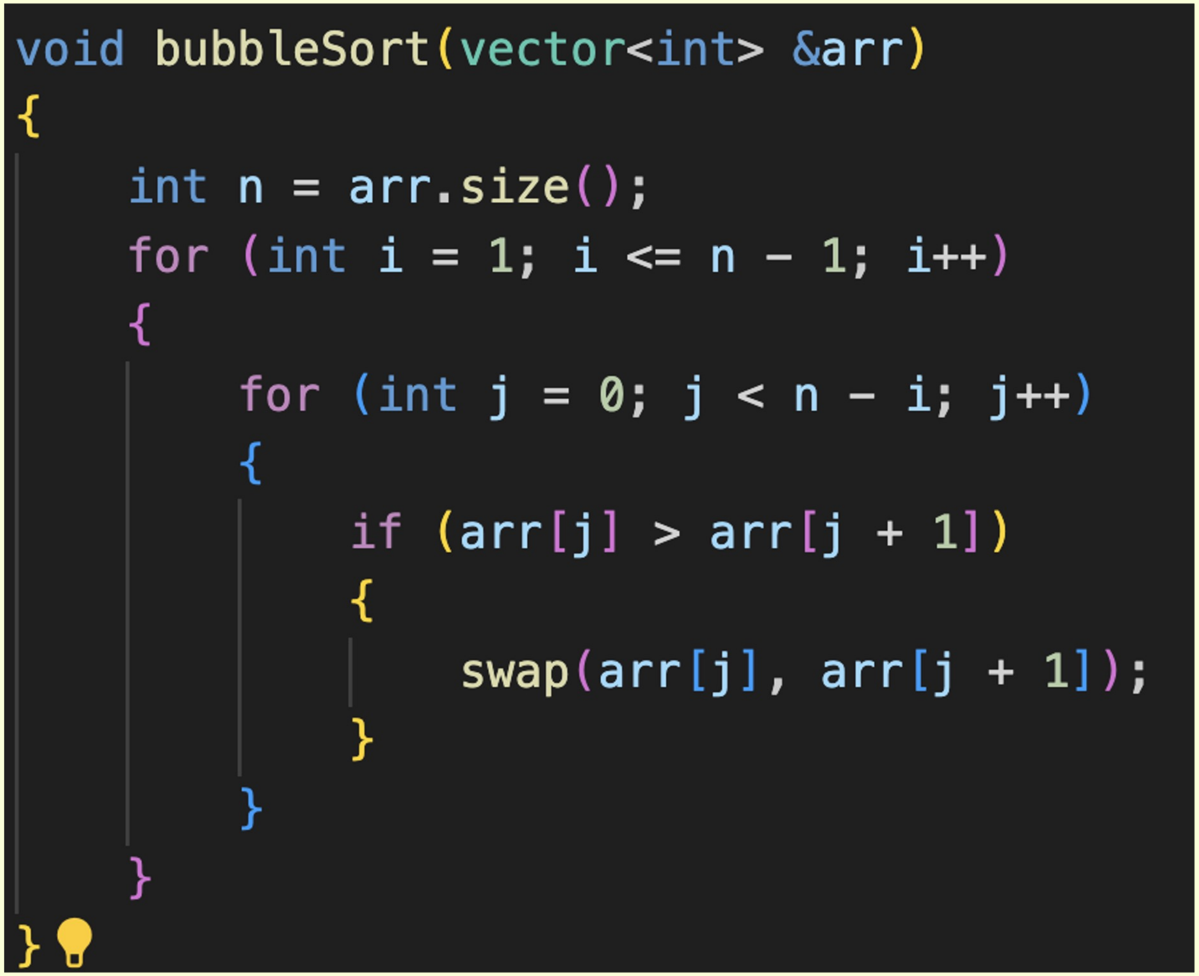
- For example:
  - If last 3 elements are bubbled up, so for the 4th bubble up we have 2 comparisons to do.



0	1	2	3	4	5
12	35	5	42	77	101

Putting It All Together

Pseudo Code



```
void bubbleSort(vector<int> &arr)
{
    int n = arr.size();
    for (int i = 1; i <= n - 1; i++)
    {
        for (int j = 0; j < n - i; j++)
        {
            if (arr[j] > arr[j + 1])
            {
                swap(arr[j], arr[j + 1]);
            }
        }
    }
}
```

Complexity Analysis

Notice that we are running the loop (N-1) times :

-For pass 1 we will have (N-1) comparisons
-For pass 2 we will have (N-2) comparisons
-For pass 3 we will have (N-3) comparisons

.
.
.
-For pass (N-1) we will have only 1 comparison

Total comparison = 1 + 2 + ………. + (N-3) + (N-2) + (N-1)

Which would be = (N-1)(N-1+1)/2 => N(N-1)/2

What do you think would be the time complexity according to the number of comparisons that we are getting?

Already Sorted Collections?

- What if the collection was already sorted?

- How many swaps will be there??

0	1	2	3	4	5
5	12	35	42	77	101

There would be 15 comparison

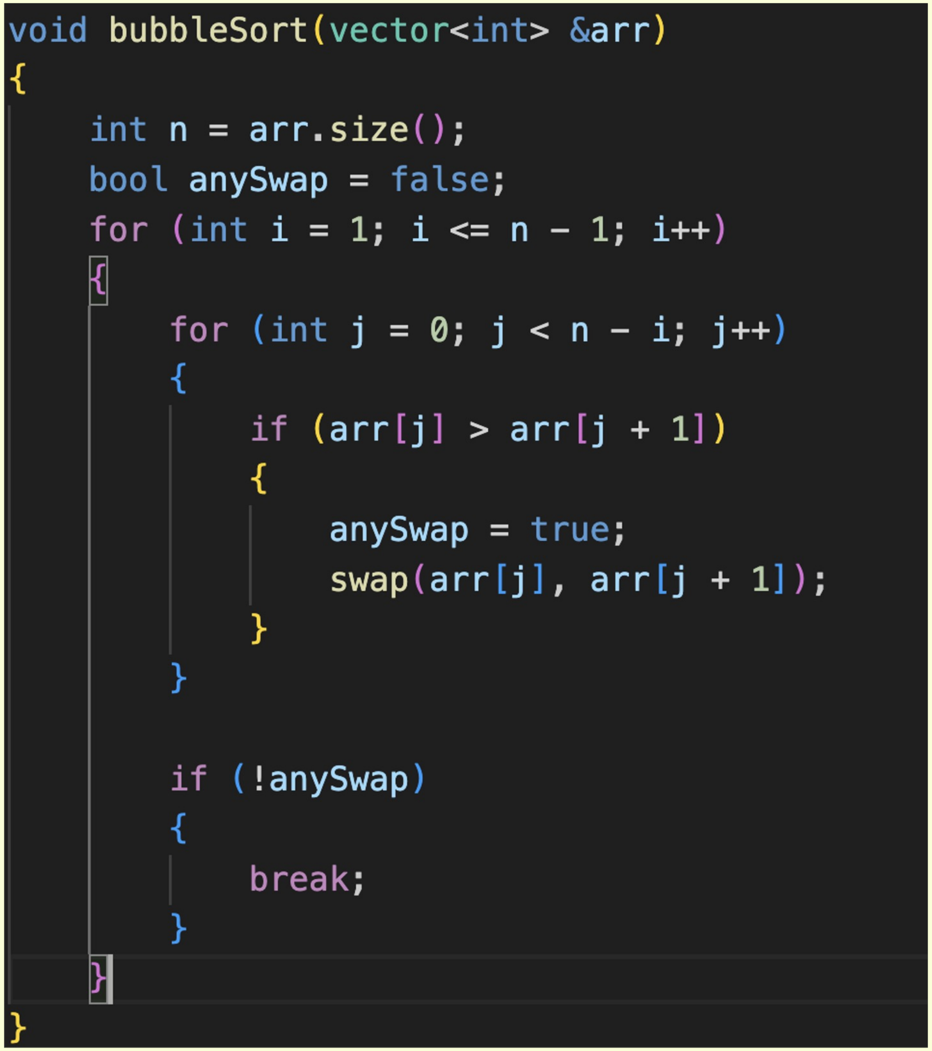
Can we reduce the number of comparison ?

Using a Boolean “Flag”

- **We can use a boolean variable to determine if any swapping occurred during the “bubble up.”**

- **If no swapping occurred, then we know that the collection is already sorted!**

Modifying Bubble Sort



```
void bubbleSort(vector<int> &arr)
{
    int n = arr.size();
    bool anySwap = false;
    for (int i = 1; i <= n - 1; i++)
    {
        for (int j = 0; j < n - i; j++)
        {
            if (arr[j] > arr[j + 1])
            {
                anySwap = true;
                swap(arr[j], arr[j + 1]);
            }
        }

        if (!anySwap)
        {
            break;
        }
    }
}
```

Can you identify the bug??

Thank you