Fundamentals of
Data Structures using C

Bubble Sort

B.Bhuvaneswaran, AP (SG) / CSE

☎ 9791519152

✉ bhuvaneswaran@rajalakshmi.edu.in

RAJALAKSHMI
ENGINEERING COLLEGE

Bubble Sort

- Bubble sort is one of the simplest internal sorting algorithms. Bubble sort works by comparing two consecutive elements and the largest element among these two bubbles towards right.
- At the end of the first pass the largest element gets sorted and placed at the end of the sorted list.
- This process is repeated for all pairs of the elements until it moves the largest element to the end of the list in that iteration.

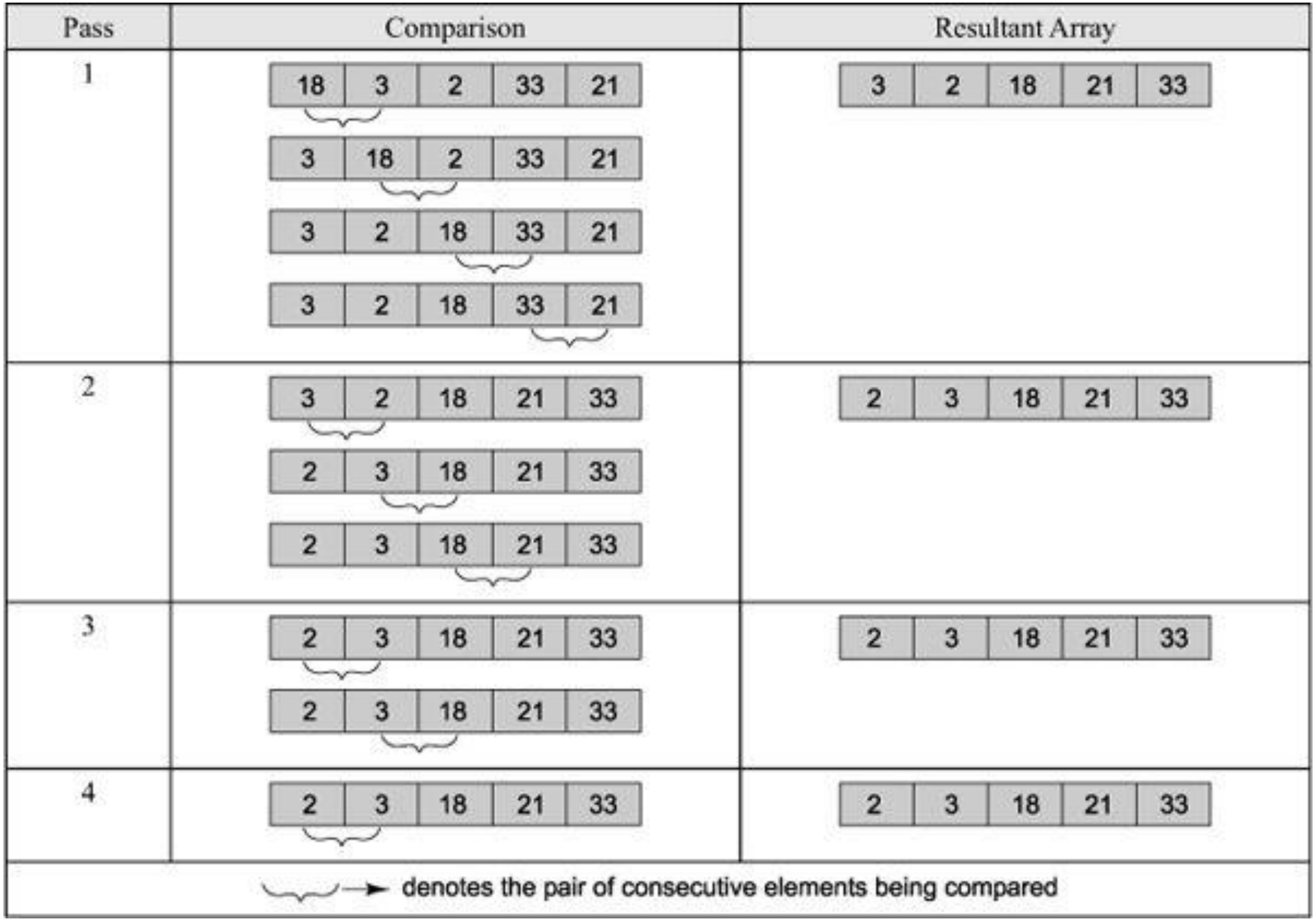
Bubble Sort

- Bubble sort consists of n-1 passes, where ‘n’ is the number of elements to be sorted.
- In the 1st pass, the largest element will be placed in the nth position.
- In the 2nd pass, the second largest element will be placed in the (n-1)th position.
- In (n-1)th pass only the first two elements are compared.

Routine

```
void BubbleSort(int a[], int n)
{
        int i, j, t;
        for (i = 0; i < n - 1; i++)
        {
                for (j = 0; j < n - 1 - i; j++)
                {
                        if (a[j] > a[j + 1])
                        {
                                t = a[j];
                                a[j] = a[j + 1];
                                a[j + 1] = t;
                        }
                }
        }
}
```

Example



Pass	Comparison	Resultant Array
1	$\underbrace{18 \quad 3} \quad 2 \quad 33 \quad 21$ $3 \quad \underbrace{18 \quad 2} \quad 33 \quad 21$ $3 \quad 2 \quad \underbrace{18 \quad 33} \quad 21$ $3 \quad 2 \quad 18 \quad \underbrace{33 \quad 21}$	3 2 18 21 33
2	$\underbrace{3 \quad 2} \quad 18 \quad 21 \quad 33$ $2 \quad \underbrace{3 \quad 18} \quad 21 \quad 33$ $2 \quad 3 \quad \underbrace{18 \quad 21} \quad 33$	2 3 18 21 33
3	$\underbrace{2 \quad 3} \quad 18 \quad 21 \quad 33$ $2 \quad \underbrace{3 \quad 18} \quad 21 \quad 33$	2 3 18 21 33
4	$\underbrace{2 \quad 3} \quad 18 \quad 21 \quad 33$	2 3 18 21 33
$\underbrace{\quad\quad}$ → denotes the pair of consecutive elements being compared		

Efficiency of Bubble Sort

- Assume that an array containing n elements is sorted using bubble sort technique.
  - Number of comparisons made in first pass = n–1.
  - Number of comparisons made in second pass = n – 2.
  - Number of comparisons made in last pass = 1.
  - Total number of comparisons made = (n – 1) + (n – 2) + . . . + 1

    = n * (n – 1) / 2

    = $O(n^2)$
- Thus, efficiency of bubble sort = $O(n^2)$.

Analysis of Bubble Sort

- Best case analysis : $O(n^2)$
- Average case analysis : $O(n^2)$
- Worst case analysis : $O(n^2)$

Advantages of Bubble Sort

- It is easy to understand and implement.
- It leverages the presence of any existing sort pattern in the list, thus resulting in better efficiency.

Limitations of Bubble Sort

- The efficiency of $O(n^2)$ is not well suited for large sized lists.
- It requires large number of elements to be shifted.
- It is slow in execution as large elements are moved towards the end of the list in a step-by-step fashion.

Queries?

Thank You!