Fundamentals of
Data Structures using C

Selection Sort

B.Bhuvaneswaran, AP (SG) / CSE

☎ 9791519152
✉ bhuvaneswaran@rajalakshmi.edu.in

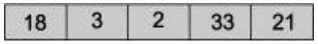
RAJALAKSHMI
ENGINEERING COLLEGE

Selection Sort

- Selection sort is one of the most basic sorting techniques.
- It works on the principle of identifying the smallest element in the list and moving it to the beginning of the list.
- This process is repeated until all the elements in the list are sorted.

Example

- Let us consider an example where a list L contains five integers stored in a random fashion, as shown:



18	3	2	33	21

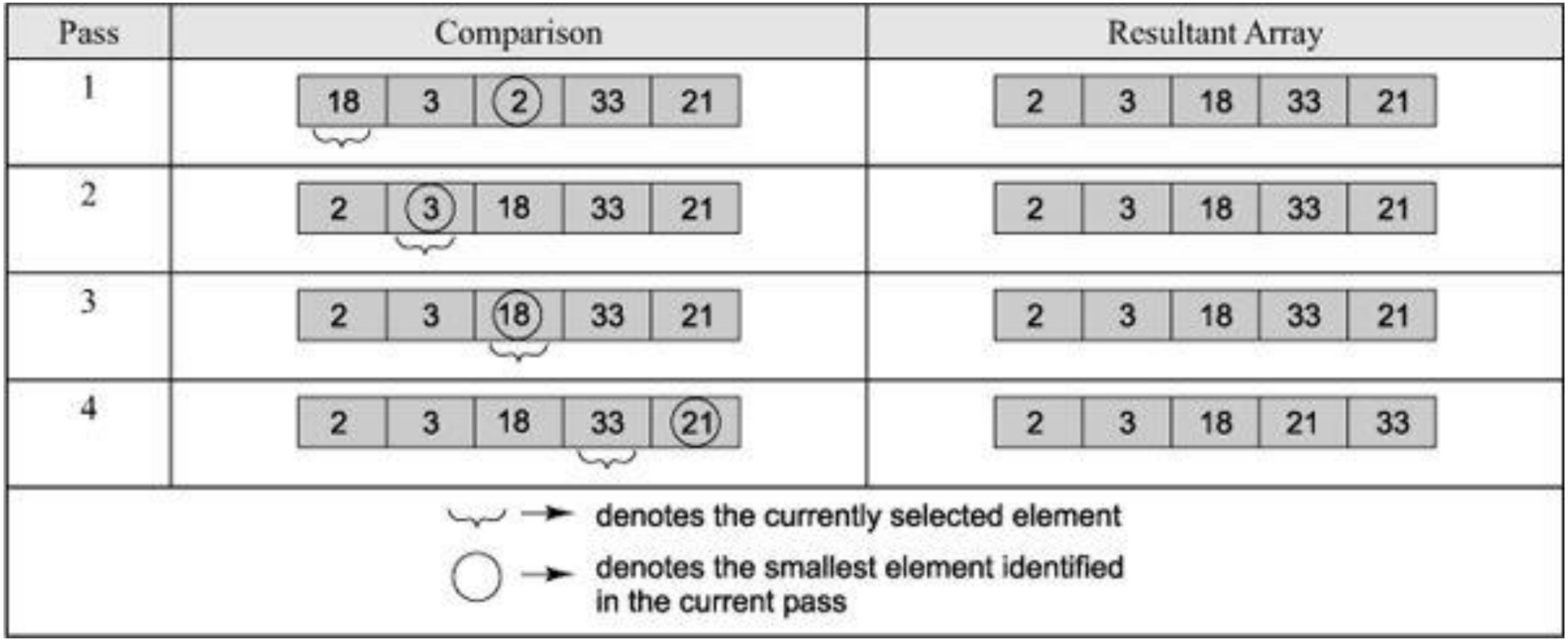
List L

- Now, if the list L is sorted using selection sort technique then first of all the first element in the list, i.e., 18 will be selected and compared with all the remaining elements in the list.
- The element which is found to be the lowest amongst the remaining set of elements will be swapped with the first element.
- Then, the second element will be selected and compared with the remaining elements in the list.
- This process is repeated until all the elements are rearranged in a sorted manner.

Passes of Selection Sort

- A single iteration of the selection sorting technique that brings the smallest element at the beginning of the list is called a pass.
- As we can see in the below, four passes were required to sort a list of five elements.
- Hence, we can say that selection sort requires n-1 passes to sort an array of n elements.

Example



Pass	Comparison					Resultant Array				
1	18	3	2	33	21	2	3	18	33	21
2	2	3	18	33	21	2	3	18	33	21
3	2	3	18	33	21	2	3	18	33	21
4	2	3	18	33	21	2	3	18	21	33

⏟ → denotes the currently selected element

○ → denotes the smallest element identified in the current pass

Routine

```
void SelectionSort(int a[], int n)
{
    int i, j, min, temp;
    for (i = 0; i < n - 1; i++)
    {
        min = i;
        for (j = i + 1; j < n; j++)
        {
            if (a[j] < a[min])
                min = j;
        }
        temp = a[i];
        a[i] = a[min];
        a[min] = temp;
    }
}
```

Efficiency of Selection Sort

- Assume that an array containing n elements is sorted using selection sort technique.
  - Now, the number of comparisons made during first pass = n – 1.
  - Number of comparisons made during second pass = n – 2.
  - Number of comparisons made during last pass = 1.
- So, total number of comparisons = (n – 1) + (n – 2) + . . . + 1

  = n * (n – 1) / 2

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

Analysis of Selection Sort

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

Advantages of Selection Sort

- It is one of the simplest of sorting techniques.
- It is easy to understand and implement.
- It performs well in case of smaller lists.
- It does not require additional memory space to perform sorting.

Limitations of Insertion Sort

- The efficiency of O(n^2) is not well suited for large sized lists.
- It does not leverage the presence of any existing sort pattern in the list.

Queries?

Thank You!