



Data Structures

Lecture 8: Sorting

Instructor:
Md Samsuddoha
Assistant Professor
Dept of CSE, BU

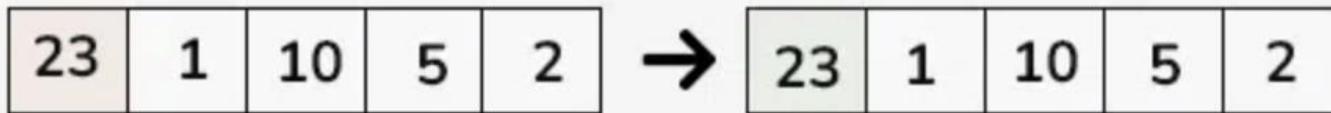
Contents

- Insertion Sort
- Counting Sort
- Merge Sort

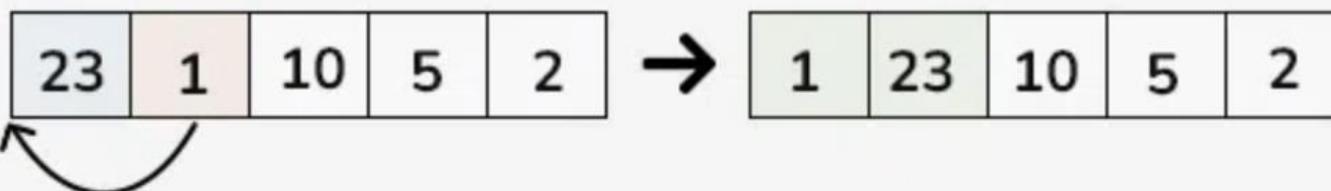
Insertion Sort

- **Insertion sort** is a simple sorting algorithm that works by iteratively inserting each element of an unsorted list into its correct position in a sorted portion of the list.
- Process of Sorting:
 - We start with the **second element** of the array as the first element is assumed to be sorted.
 - Compare the second element with the first element if the second element is smaller then swap them.
 - Move to the third element, compare it with the first two elements, and put it in its correct position
 - Repeat until the entire array is sorted.

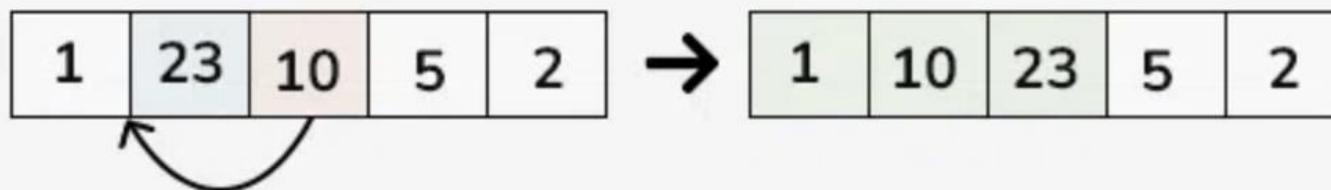
Initially



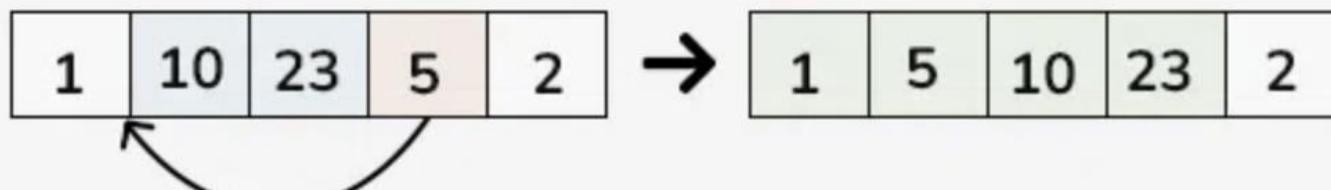
First Pass



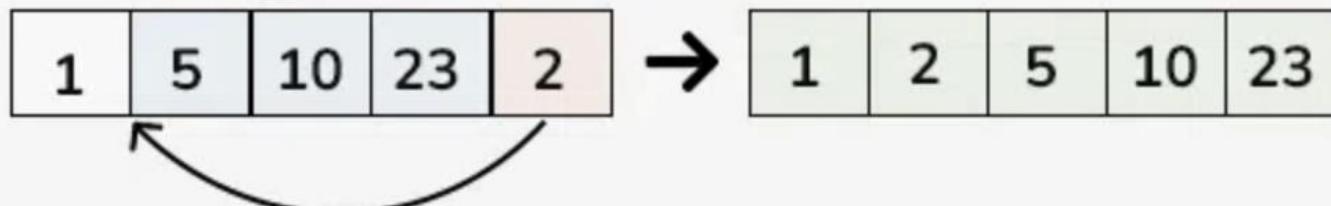
Second Pass



Third Pass



Fourth Pass



Complexity

- Worst case: array is in **reverse order**.
- Number of Iterations: N-1
- Number of comparisons in each pass:
 - 1st pass → 1 comparison
 - 2nd pass → 2 comparisons
 - ...
 - Last pass → n-1 comparisons
- Total comparisons = $1+2+3 \dots + (n-1) = n(n-1)/2$
- **Worst-case time complexity:** $O(n^2)$

Pseudocode

```
InsertionSort(A, n) :  
    for i from 1 to n-1:  
        key = A[i]  
        j = i - 1  
  
        while j >= 0 and A[j] > key:  
            A[j + 1] = A[j]  
            j = j - 1  
  
        A[j + 1] = key
```

Coding

- Write a C program for Insertion sort.

References

- **Chapter 10: Data Structures using C** by E. Balagurusamy
- Visit the site for live visualization: <https://visualgo.net/>

Thank You