SELECTION SORT

1. Definition

Selection Sort is a comparison-based sorting algorithm that repeatedly selects the **minimum element** from the unsorted portion of the array and places it in its correct position in the sorted portion.

2. Key Idea

1. Divide the array into **sorted** and **unsorted** regions.
2. Find the **minimum** element in the unsorted region.
3. Swap it with the **first element** of the unsorted region.
4. Expand the sorted region by one and reduce the unsorted region.
5. Repeat until the entire array is sorted.

3. Example

Array: `[5, 4, 3, 2, 1]`

Pass 1

Unsorted part: `[5, 4, 3, 2, 1]`
Minimum = 1 → swap with 5
Array → `[1, 4, 3, 2, 5]`

Pass 2

Unsorted part: `[4, 3, 2, 5]`
Minimum = 2 → swap with 4
Array → `[1, 2, 3, 4, 5]`

Pass 3

Minimum = 3 → swap with 3
Array unchanged

Pass 4

Minimum = 4 → swap with 4

4. Algorithm (Pseudocode)

```
SelectionSort(arr, n)
    for i = 0 to n-2
        minIndex = i
        for j = i+1 to n-1
            if arr[j] < arr[minIndex]
                minIndex = j
        swap arr[i], arr[minIndex]
```

5. Properties

Property	Value
Algorithm Type	Comparison-based
Technique	Selection
In-place	Yes
Stable	❌ No (can be made stable with extra work)
Adaptive	❌ No (does the same work even if the array is sorted)

8. Advantages

- Easy to understand
- Fewer swaps than bubble sort
- Good when the wap cost is expensive
- Works well for small datasets

9. Disadvantages

- Very slow for large inputs
- Always $O(n^2)$, even if the array is sorted
- Not stable
- Not adaptive

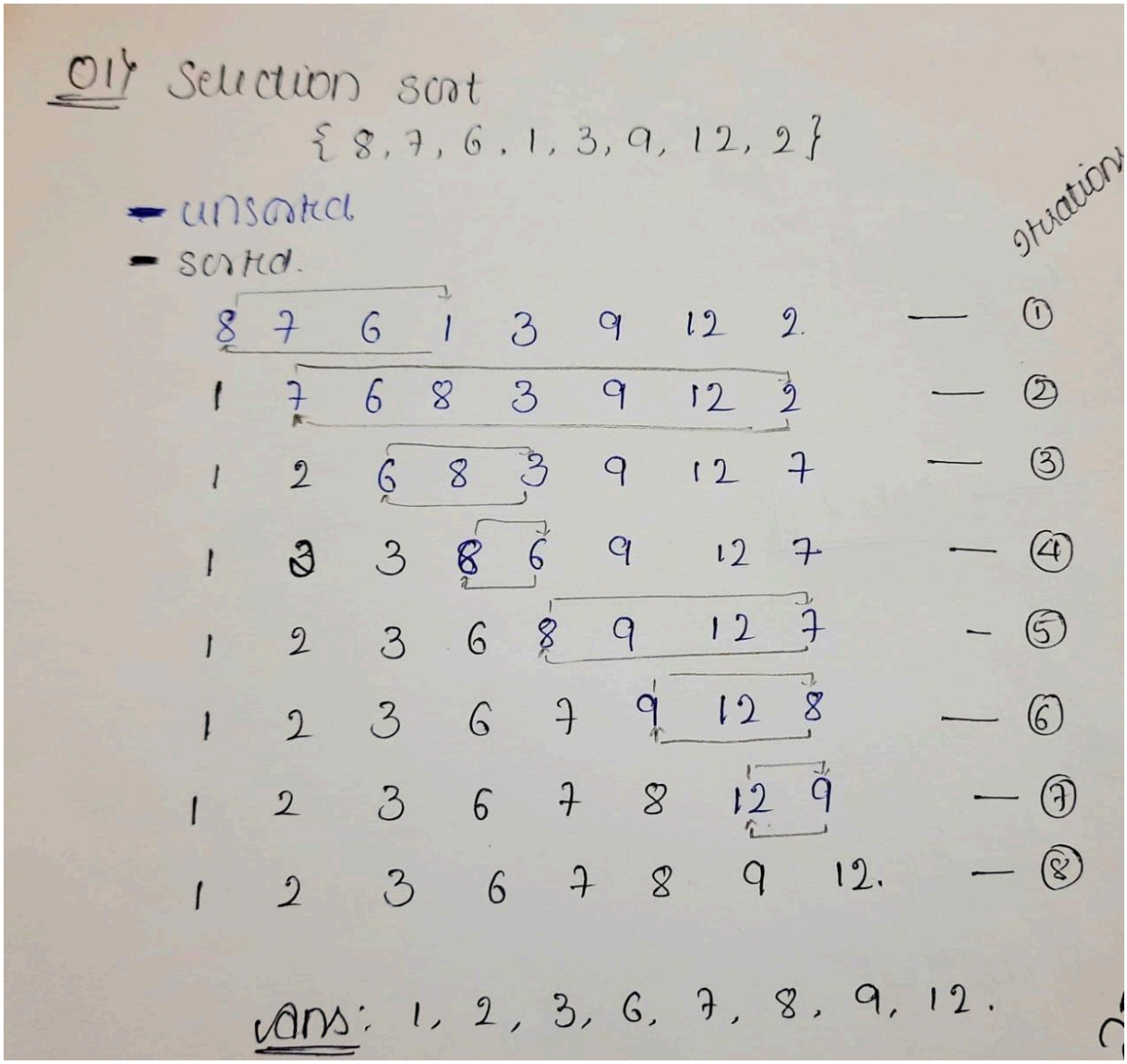
10. Real-Life Analogy

Imagine selecting the **smallest shirt** from a pile of clothes, placing it on top, then selecting the next smallest, and so on.

11. Exam Notes

- Selection sort repeatedly selects minimum from unsorted region.
- Always makes **$n(n-1)/2$ comparisons**.
- Worst, best, average = **$O(n^2)$**.
- Only **n – 1 swaps**, making it efficient for costly swaps.
- Not stable, not adaptive, in-place algorithm.

12. EFFICIENCY ANALYSIS IS THE SAME AS BUBBLE SORT


01) Selection sort
{8, 7, 6, 1, 3, 9, 12, 2}
- unsorted
- sorted.
Situation
8 7 6 1 3 9 12 2. — ①
1 7 6 8 3 9 12 2 — ②
1 2 6 8 3 9 12 7 — ③
1 2 3 8 6 9 12 7 — ④
1 2 3 6 8 9 12 7 — ⑤
1 2 3 6 7 9 12 8 — ⑥
1 2 3 6 7 8 12 9 — ⑦
1 2 3 6 7 8 9 12. — ⑧
Ans: 1, 2, 3, 6, 7, 8, 9, 12.