ETHNUS ™

Explore | Expand | Enrich

SORTING
TECHNIQUES
IN
JAVA

- A Sorting Algorithm is used to rearrange a given array or list elements according to a comparison operator on the elements

- The comparison operator is used to decide the new order of element in the respective data structure

- It is the simplest sort method which performs sorting by repeatedly moving the largest element to the highest index of the array

- It comprises of comparing each element to its adjacent element and replace them accordingly
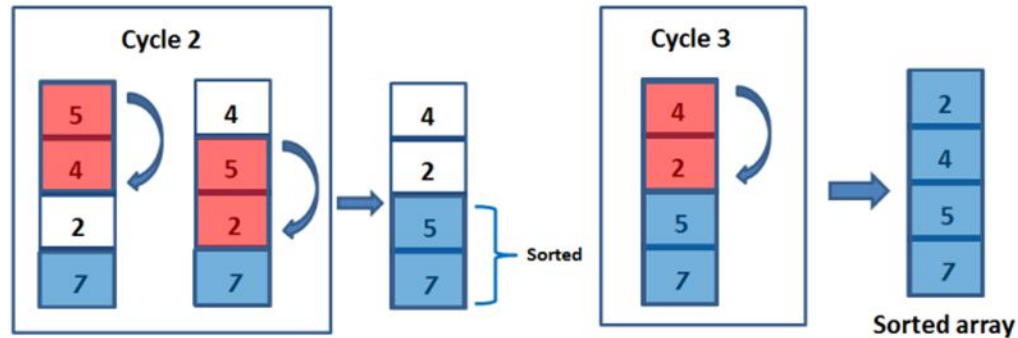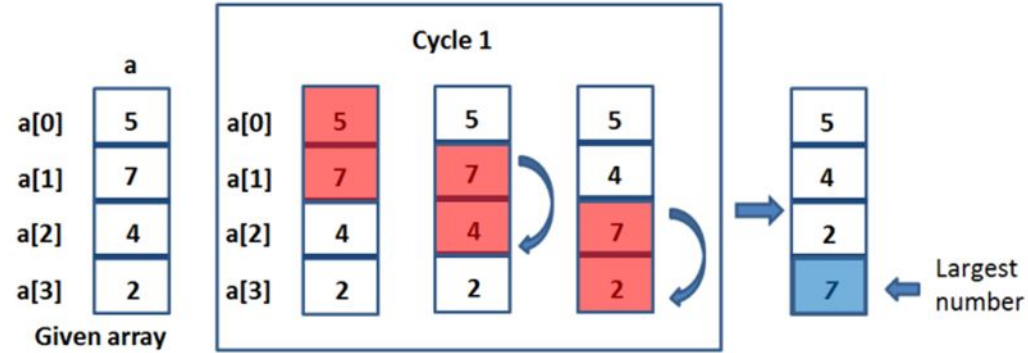
- Starting with the first element(index = 0), compare the current element with the next element of the array

- If the current element is greater than the next element of the array, swap them

- If the current element is less than the next element, move to the next element. Repeat Step 1

Bubble Sort

```
class BubbleSort {
    void bubbleSort(int arr[]) {
    int n = arr.length;
    for (int i = 0; i < n - 1; i++)
        for (int j = 0; j < n - i - 1; j++)
            if (arr[j] > arr[j + 1]) {
                // swap temp and arr[i]
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
    }
```
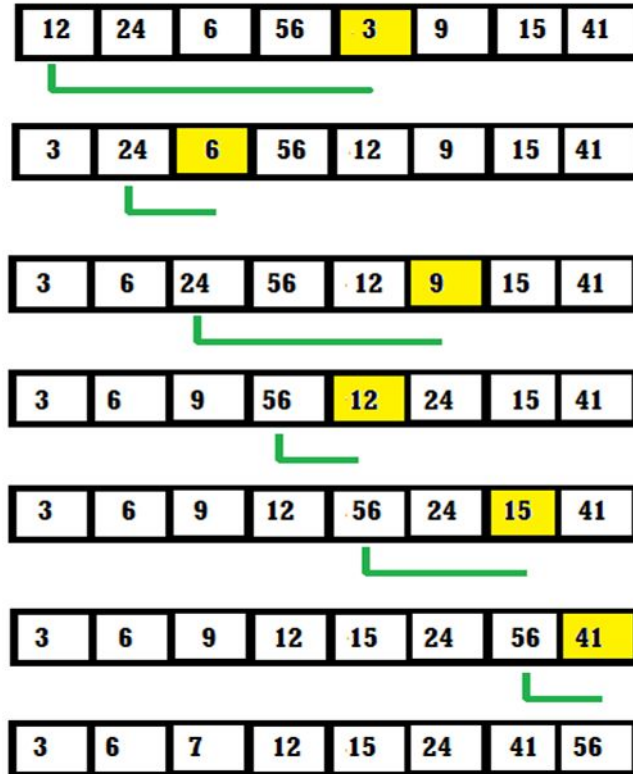
- Find the minimum element in the list

- Swap it with the element in the first position of the list

- Repeat the steps above for all remaining elements of the list starting from the second position

| 12 | 24 | 6 | 56 | **3** | 9 | 15 | 41 |

| 3 | 24 | **6** | 56 | 12 | 9 | 15 | 41 |

| 3 | 6 | 24 | 56 | 12 | **9** | 15 | 41 |

| 3 | 6 | 9 | 56 | **12** | 24 | 15 | 41 |

| 3 | 6 | 9 | 12 | 56 | 24 | **15** | 41 |

| 3 | 6 | 9 | 12 | 15 | 24 | 56 | **41** |

| 3 | 6 | 7 | 12 | 15 | 24 | 41 | 56 |

```java
class SelectionSort {
    void sort(int arr[]) {
    int n = arr.length;

    // One by one move boundary of
unsorted subarray
    for (int i = 0; i < n - 1; i++) {
        // Find the minimum element in
unsorted array
        int min_idx = i;
        for (int j = i + 1; j < n;
j++)
            if (arr[j] <
arr[min_idx])
                min_idx = j;

            // Swap the found minimum
element with the first
            // element
            int temp = arr[min_idx];
            arr[min_idx] = arr[i];
            arr[i] = temp;
    }
    }
```

THANK YOU