

ALGORITHMS AND DATA STRUCTURES

CS106.3

1. Explain how main sorting algorithms can be performed using an appropriate array example

- Selection sort:
 - ❖ Selection sort divides the array into two parts as sorted part and unsorted part. In each pass, the smallest element from the unsorted part is selected and swap with the first element of the unsorted part.

Assume we have an array of integers.

5	2	8	1
---	---	---	---



Sorted part



Sorted part



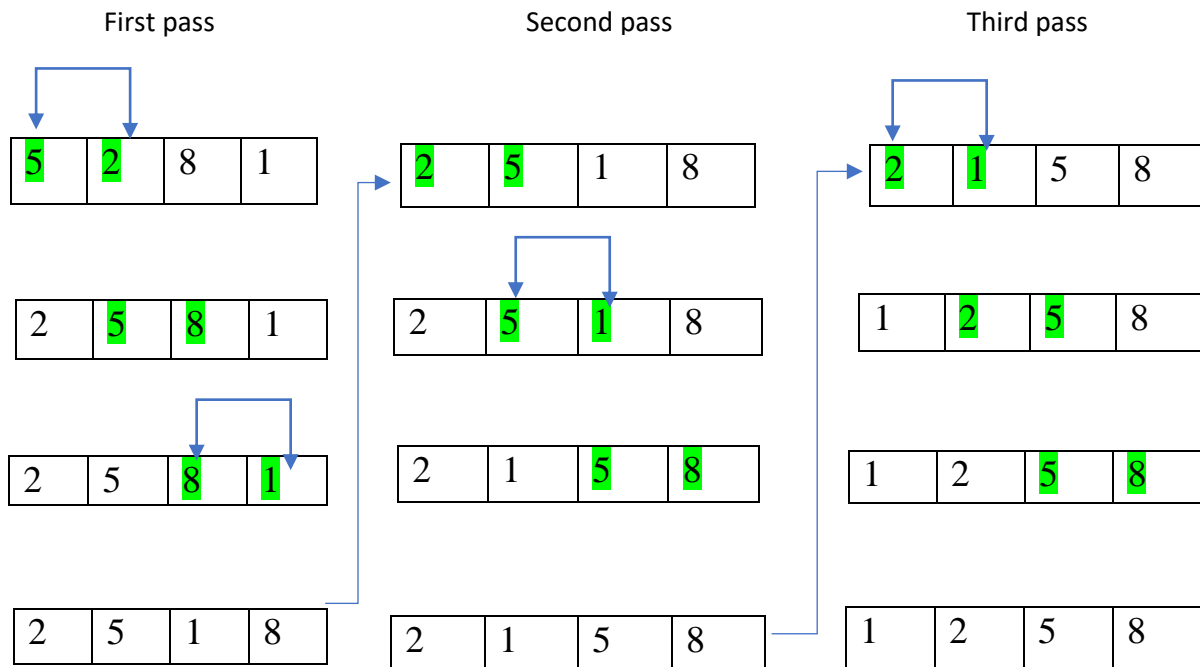
Sorted part



Sorted array

- Bubble sort:

- ❖ In each pass of bubble sort, adjacent elements are compared and swapped if they are in the wrong order. The largest element “bubbles” to the end of the array in each pass.



2. Compare and contrast bubble sort and selection sort algorithm

- Bubble sort:

- It compares adjacent elements and swaps them if they are in wrong order
- It repeatedly bubbles the largest element to the end of the array
- It has time complexity $O(n^2)$ in the worst case

- Selection sort:

- It divides the array into two parts as sorted part and unsorted part.
- It repeatedly selects the smallest element from the unsorted part and swaps it with the first element of the unsorted part
- It has time complexity $O(n^2)$ in the worst case

3. What are the real-world examples for sorting

- Sorting a list of names alphabetically in a phonebook or contact list
- Sorting a list of products based on their prices for e-commerce websites
- Sorting students' grades from highest to lowest to determine their ranking
- Sorting files by date modified to locate the most recent one

4. Write a function using pseudocode or source codes to sort an integer array using bubble sort and selection sort

Selection sort:

```
void selectionSort(int array[], int size)
{
    int startScan, minIndex, minValue;

    for (startScan = 0; startScan < (size - 1); startScan++)
    {
        minIndex = startScan;
        minValue = array[startScan];
        for(int index = startScan + 1; index < size; index++)
        {
            if (array[index] < minValue)
            {
                minValue = array[index];
                minIndex = index;
            }
        }
        array[minIndex] = array[startScan];
        array[startScan] = minValue;
    }
}
```

Bubble sort:

```
do {  
    swp=false;  
    // different passes  
    for(j=0 ; j< n-1 ; j++)  
    {  
        // swap if incorrect order  
        if( arr[j]> arr[j+1])  
        {  
            temp=arr[j];  
            arr[j]=arr[j+1];  
            arr[j+1]=temp;  
            swp=true;  
        }  
    }  
} while(swp);
```