

Tutorial 05 Sorting Algorithms

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

Start with the following array:

[7, 2, 9, 1, 5]

- a) Bubble Sort: Bubble sort works by repeatedly swapping adjacent elements if they are in the wrong order. It continues until the array is sorted. Here's how it works:

Pass 1 : [2, 7, 1, 5, 9]

Pass 2 : [2, 1, 5, 7, 9]

Pass 3 : [1, 2, 5, 7, 9]

Final Sorted Array: [1, 2, 5, 7, 9]

- b) Selection Sort: Selection sort finds the minimum element from the unsorted portion of the array and swaps it with the element at the beginning of the unsorted portion. This process is repeated until the array is sorted.

Pass 1 : [1, 2, 9, 7, 5]

Pass 2 : [1, 2, 9, 7, 5]

Pass 3 : [1, 2, 5, 7, 9]

Final Sorted Array: [1, 2, 5, 7, 9]

- (2) Compare and contrast bubble sort and selection sort algorithms.

Bubble Sort

Bubble sort is a simple sorting algorithm which continuously moves through the list and compares adjacent pairs for proper sorting.

Selection Sort

Selection sort is a sorting algorithm which takes either smallest value or largest value in the list and place it at the proper position in the list.

of the elements.	
Bubble sort compares the adjacent elements and move accordingly.	Selection sort selects the smallest element from the unsorted list and moves it at the next position of the sorted list.
Bubble sort performs a large number of swaps or moves to sort the list.	Selection sort performs comparatively less number of swaps or moves to sort the list.
Bubble sort is relatively slower.	Selection sort is faster as compared to bubble sort.
The efficiency of the bubble sort is less.	The efficiency of the selection sort is high.
Bubble sort performs sorting of an array by exchanging elements.	Selection sort performs sorting of a list by the selection of element.

Selection sort and bubble sort also has a space complexity of $O(1)$.

- (Q) What are the real-world example of sorting
- Phonebook or Contact list
 - Online Marketplaces
 - Filesystems
 - Library Catalogs
 - Music Playlists
 - Search Engine Result Pages
 - Routing Algorithms
 - Event Scheduling
 - Leaderboards and Rankings
 - Address Verifications

(1) Write a function using pseudo or source codes to sort an integer array using bubble sort and selection sort.

```
#include <stdio.h>
void bubbleSort(int array[], int size){
    for(int step = 0; step < size - 1; ++step) {
        for(int i = 0; i < size - step - 1; ++i) {
            if(array[i] > array[i + 1]) {
                int temp = array[i];
                array[i] = array[i + 1];
                array[i + 1] = temp;
            }
        }
    }
}
```

```
void printArray(int array[], int size) {
    for(int i = 0; i < size; ++i) {
        printf("%d", array[i]);
    }
    printf("\n");
}
```

```
int main() {
    int data[] = {2, 45, 0, 11, 9};
    int size = 5;
    bubbleSort(data, size);
}
```

```
printf("Sorted Array in Ascending order: \n");
printArray(data, size);
}
```

```
#include <stdio.h>
void selectionSort(int array[], int size){
    for(int step = 0; step < size - 1; step++){
        int min_idx = step;
        for(int i = step + 1; i < size; i++){
            if(array[i] < array[min_idx])
                min_idx = i;
        }
        swap(&array[min_idx], &array[step]);
    }
}

void printArray(int array[], int size){
    for(int i = 0; i < size; ++i){
        printf("%d ", array[i]);
    }
    printf("\n");
}

int main(){
    int data[] = {20, 12, 10, 15, 2};
    int size = 5;
    selectionSort(data, size);
    printf("Sorted array in Ascending Order:\n");
    printArray(data, size);
}
```