



HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

# C Programming Basic

## Sorting - part 1

# CONTENT

---

- Data generation
- Implement selection sort, insertion sort, and bubble sort algorithms
- Experiments

# Exercise 1: data generation

---

- Write a program that inputs 3 positive integers  $n$ ,  $m$ ,  $M$ , and generates  $n$  positive integers  $a_1, a_2, \dots, a_n$  which are random from  $m$  to  $M$ . Store the sequence to a file arr-n.txt with following format:
  - Line 1: a positive integer  $n$  ( $1 \leq n \leq 10^6$ )
  - Line 2: write  $a_1, a_2, \dots, a_n$  separated by a SPACE character

# Insertion Sort

Algorithm:

- The array is divided into two subarrays, *sorted* and *unsorted*
  - Each iteration: the first element of the unsorted subarray is picked up, transferred to the sorted subarray, and inserted at the appropriate place.
- ➔ An array of  $n$  elements requires  $n-1$  iterations to completely sort the array.

Sorted	Unsorted					
23	78	45	8	32	56	Original array
23	78	45	8	32	56	After iteration 1
23	45	78	8	32	56	After iteration 2
8	23	45	78	32	56	After iteration 3
8	23	32	45	78	56	After iteration 4
8	23	32	45	56	78	After iteration 5

# Insertion Sort

- Running time
  - Worst case:  $O(n^2)$
  - Best case:  $O(n)$

```
void insertionSort(int A[], int N) {  
    // index tu 1 -> N  
    for(int k = 2; k <= N; k++){  
        int last = A[k];  
        int j = k;  
        while(j > 1 && A[j-1] >  
            last){  
            A[j] = A[j-1];  
            j--;  
        }  
        A[j] = last;  
    }  
}
```

# Exercise 2

- Write a program that inputs a sequence of positive integers  $a_1, a_2, \dots, a_n$ , sort the given sequence in non-decreasing order by the insertion sort algorithm
- Input (stdin)
  - Line 1: a positive integer  $n$  ( $1 \leq n \leq 10^6$ )
  - Line 2: integers  $a_1, a_2, \dots, a_n$ , ( $1 \leq a_i \leq 10^6$ )
- Result (stdout)
  - Write the sorted sequence, elements are separated by a SPACE character

stdin	stdout
4 5 4 3 4 1 2	1 2 3 4 4 5

# Selection Sort

---

- The array is divided into two subarrays, *sorted* and *unsorted*, which are divided by an imaginary wall.
- Each iteration: We find the smallest element from the unsorted subarray and swap it with the element at the beginning of the unsorted subarray
- ➔ After each iteration (after each selection and swapping): the imaginary wall between the two subarrays move one element ahead, increasing the number of sorted elements and decreasing the number of unsorted ones.
- An array of  $n$  elements requires  $n-1$  iterations to completely sort the array.

# Selection Sort

- Running time
  - Worst case:  $O(n^2)$
  - Best case:  $O(n^2)$

```
void selectionSort(int A[], int N) {  
    // index tu 1 -> N  
    for(int k = 1; k <= N; k++){  
        int min = k;  
        for(int j = k+1; j <= N; j++){  
            if(A[min] > A[j]) min = j;  
        }  
        int tmp = A[min];  
        A[min] = A[k];  
        A[k] = tmp;  
    }  
}
```



# Selection Sort

- Example: 5, 7, 3, 8, 1, 2, 9, 4, 6

5	7	3	8	1	2	9	4	6
---	---	---	---	---	---	---	---	---



1	7	3	8	5	2	9	4	6
---	---	---	---	---	---	---	---	---



1	2	3	8	5	7	9	4	6
---	---	---	---	---	---	---	---	---



1	2	3	8	5	7	9	4	6
---	---	---	---	---	---	---	---	---



1	2	3	4	5	7	9	8	6
---	---	---	---	---	---	---	---	---



1	2	3	4	5	7	9	8	6
---	---	---	---	---	---	---	---	---



1	2	3	4	5	6	9	8	7
---	---	---	---	---	---	---	---	---



1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---



# Exercise 3

- Write a program that inputs a sequence of positive integers  $a_1, a_2, \dots, a_n$ , sort the given sequence in non-decreasing order by the selection sort algorithm
- Input (stdin)
  - Line 1: a positive integer  $n$  ( $1 \leq n \leq 10^6$ )
  - Line 2: integers  $a_1, a_2, \dots, a_n$ , ( $1 \leq a_i \leq 10^6$ )
- Result (stdout)
  - Write the sorted sequence, elements are separated by a SPACE character

stdin	stdout
4 5 4 3 4 1 2	1 2 3 4 4 5

# Bubble Sort

- Traverse the sequence from left to right
  - Swap two adjacent elements if they are not in the right order
- Repeat that traversal if the previous step has swaps
- Running time
  - Worst case:  $O(n^2)$
  - Best case:  $O(n)$

```
void bubbleSort(int A[], int N) {  
    // index tu 1 den N  
    int swapped;  
    do{  
        swapped = 0;  
        for(int i = 1; i < N; i++){  
            if(A[i] > A[i+1]){  
                int tmp = A[i];  
                A[i] = A[i+1];  
                A[i+1] = tmp;  
                swapped = 1;  
            }  
        }  
    }while(swapped == 1);  
}
```

# Bubble Sort

- Example: 5, 7, 3, 8, 1, 2, 9, 4, 6

5	3	7	1	2	8	4	6	9
---	---	---	---	---	---	---	---	---

3	5	1	2	7	4	6	8	9
---	---	---	---	---	---	---	---	---

3	1	2	5	4	6	7	8	9
---	---	---	---	---	---	---	---	---

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

# Exercise 4

- Write a program that inputs a sequence of positive integers  $a_1, a_2, \dots, a_n$ , sort the given sequence in non-decreasing order by the bubble sort algorithm
- Input (stdin)
  - Line 1: a positive integer  $n$  ( $1 \leq n \leq 10^6$ )
  - Line 2: integers  $a_1, a_2, \dots, a_n$ , ( $1 \leq a_i \leq 10^6$ )
- Result (stdout)
  - Write the sorted sequence, elements are separated by a SPACE character

stdin	stdout
4 5 4 3 4 1 2	1 2 3 4 4 5



25 YEARS ANNIVERSARY  
**SOICT**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

