Assignment 6 – Sorting Algorithms

TA: Helen (j15935741@gmail.com)
Deadline: Jan. 11, 11:59pm

1. Implementation of sorting algorithms

Sorting is a common and important problem in programming. To understand the different sorting algorithms, we are going to implement them on our own. Before we start implementing the sorting algorithms, we need an array containing integers. You should read a “test.txt” file which contains unsorted integers listed as follows.

```
41 24 76 11 45 64 21 69 19 36
```

This is the array with the integers we want to test our sorting algorithms. There are many sorting algorithms and for this assignment, please implement four sorting algorithms.

The numbers in the output array should be in the following format

```
for(i=0;i<array_lengh;i++){
    printf("%3d ",array[i]);
}
```

1.1 Print input data 10%

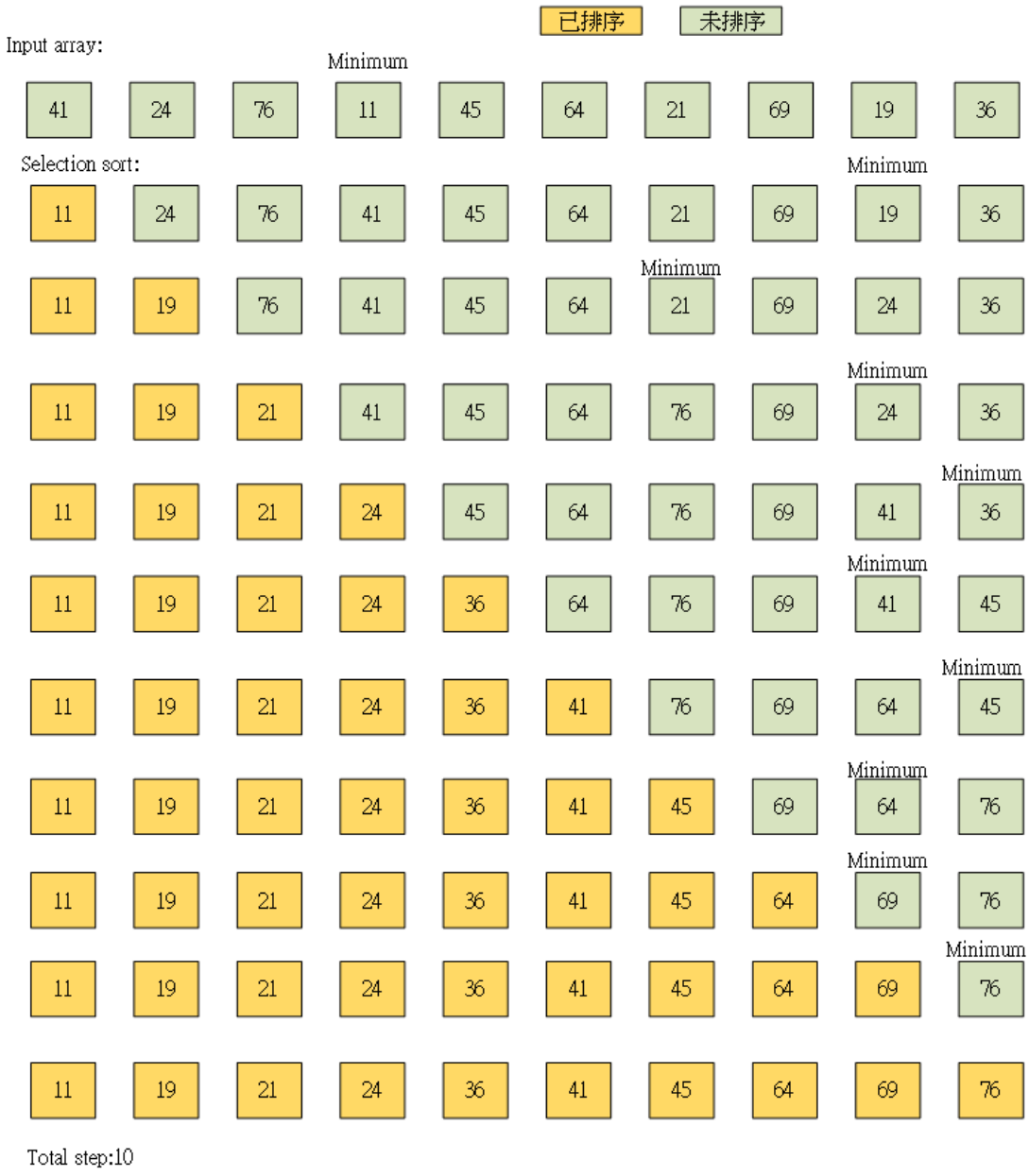
Before implementing the following algorithms you should print out your array. The output format is shown below:

```
Input array:
 41  24  76  11  45  64  21  69  19  36
```

1.2 Selection Sort 20%

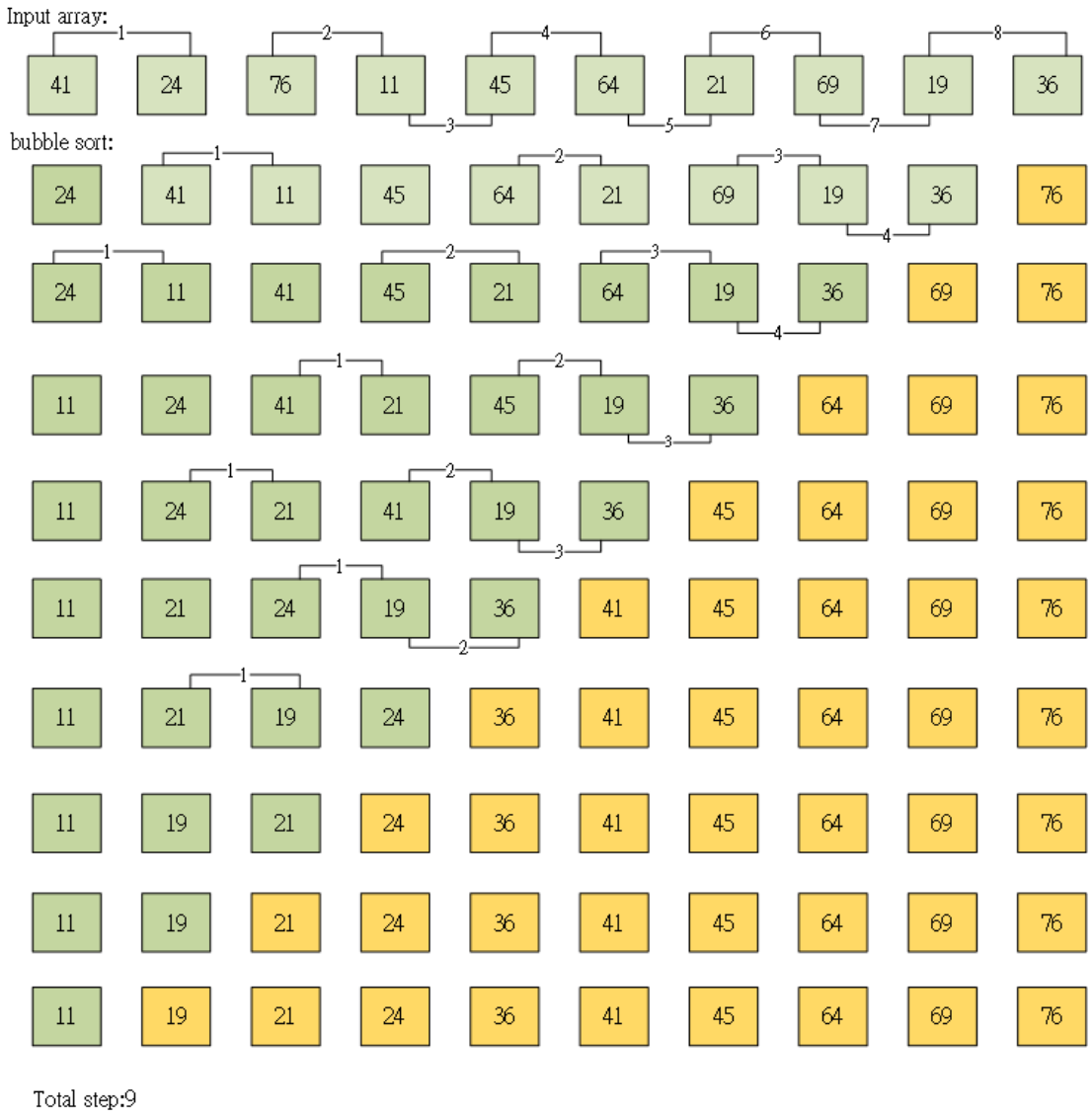
Please implement your own selection_sort() function. The selection sort algorithm searches the whole array for the smallest element.

First, the data is divided into two parts, sorted and unsorted. Suppose you want to put n data from small to large, and the n data at the beginning is "unsorted". At this time, the minimum value of n "unsorted" data is taken out and placed at the end of the "sorted" data, that is, it is exchanged with the first data, and the array result is printed. And so on, until there is no "unsorted" data, the procedure stops. Finally, you must print how many steps in this algorithm. The format of the output is as follows.



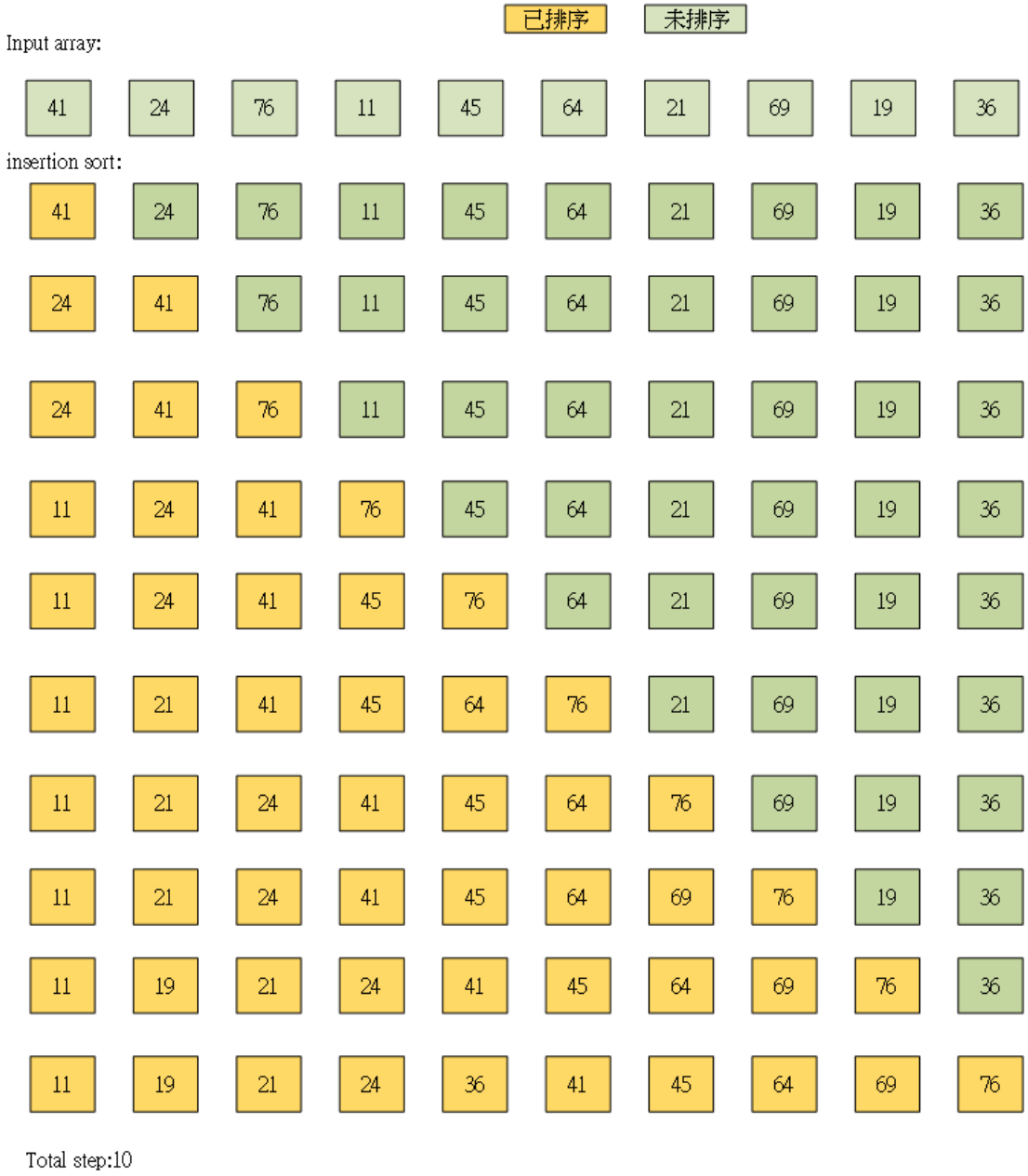
1.3 Bubble sort 20%

Please implement the bubble sort function: bubble_sort(). Starting from the front of the array, compare the two neighbor elements in the array at a time, then change them according to the size, and move them to the back. When we compare all the elements once, we can ensure that the element with the largest value is at the end. And print out the array results, until the entire array is sorted. Finally, you must print how many steps in this algorithm. The format of the output is as follows.



1.4 Insertion sort 20%

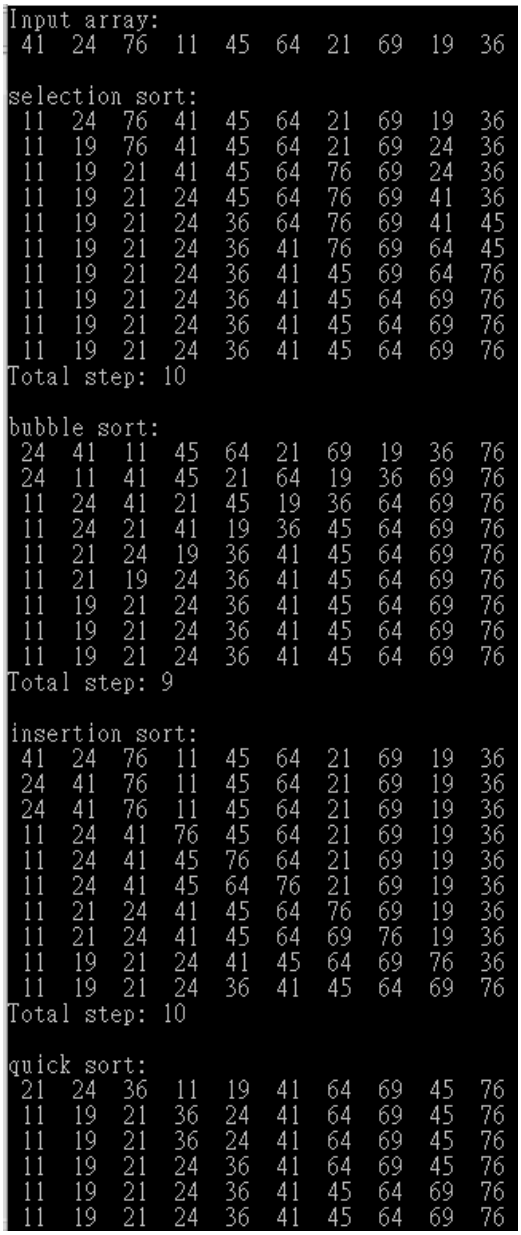
Please implement the insertion sort function insertion_sort(). Insertion sort divides the data into sorted, unsorted parts, and takes an element from the top of the unsorted array. From the back to the front and the sorted array, until the value is not bigger than the element and after inserting this value; if none of them are inserted in the front, and then print out the array results, repeat the above actions until the unsorted series are all processed. Finally, you must print how many steps in this algorithm. The format of the output is as follows.

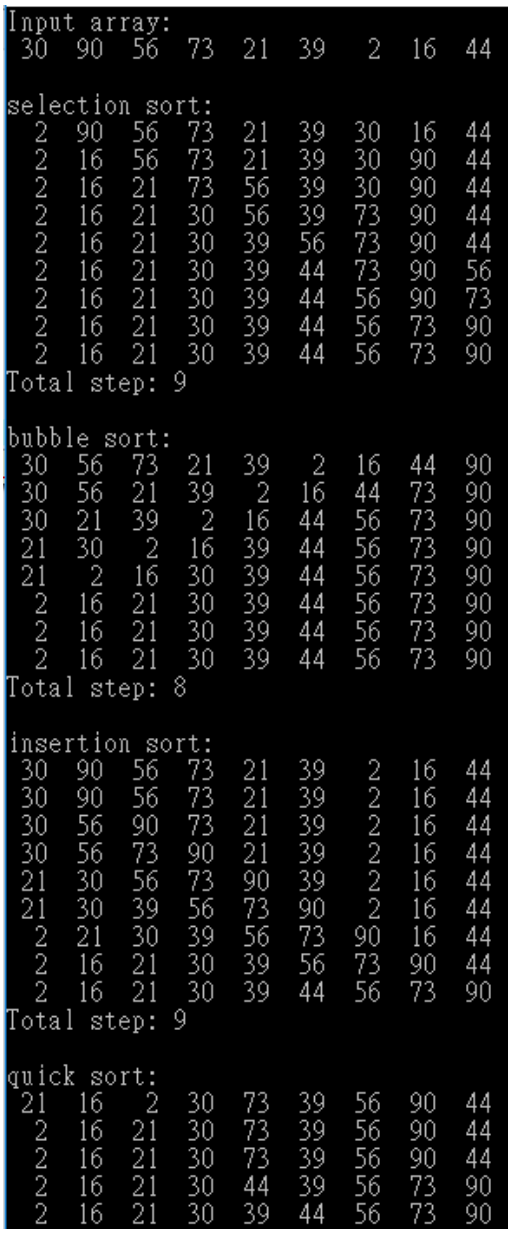


1.5 Quick Sort 25%

Please implement the quick sort function quick_sort(). The quick sort algorithm uses a divide and conquer strategy. **It selects the first number as the pivot element** and moves all the smaller elements to the left of the array, moves all the elements to the larger right side, and prints the result. It will recursively perform this operation on the left and right parts of the array, specifying that you should complete the left part and then until the whole array is sorted. The format of the output is as follows.

2. Test case (Input data will not be the same as test case when demo)





3. Output, readme, comments and style (5%)

The TA(s) will mark and give points according to the following:

- 10% print input data
- 20% selecton sort
- 20% bubbe sort
- 20% insertion sort
- 25% quick sort
- 5% Readme file, code style, and comments in source code

Readme file should include your name, class ID, a brief description of the code, and other issues students think that will be helpful for the TAs to understand their homework.

4. How to submit

To sbmit your files electronically, enter the following command from the csie workstation:

turnin DS_I_2018.hw6 [your files…]

To check the files you turnin, enter the following command from the csie workstation:

turnin –ls DS_I_2018

You can see other description about turnin from following link:
https://www.cs.ccu.edu.tw/lab401/doku.php?id=turninhowto