Department of Computer Science and Engineering
National Sun Yat-sen University
Data Structures Quiz, Chapter 7, Nov. 25, 2024

1. (a) Please explain the *insertion sort* algorithm using the input elements 5, 9, 2, 8, 6, sorting them in nondecreasing order. (15%)

   (b) The following is an implementation of the *insertion sort* algorithm. Please explain the purpose of Line 6. (15%)

```
1: void InsertionSort(int a[ ], int n)
2: // Sort a[1], a[2], ...a[n] into nondecreasing order
3: {
4:       for (int j = 2; j <= n; j++)
5:       {
6:            a[0] = a[j];
7:            int i = j-1;    int e = a[j];
8:            while (e < a[i])
9:            {
10:                a[i+1] = a[i];
11:                i--;
12:           }
13:           a[i+1] = e;
14:      }
15:}
```

2. (a) Explain the *quick sort* method with the input: 26, 5, 37, 1, 61, 11, 59, 15, 48, 19. (15%)

   (b) Please present a mathematical analysis of the time complexity for the best-case of the *quick sort* method. (15%)

3. Write a recursive C++ function to perform the *recursive merge sort*. To implement your merge sort, you can call the following 2-way merge function as a basic function to merge two sorted arrays into a single one. There is no need to write the body of the 2-way merge function. (40%)

```
void twoway(int a[ ], int b[ ], int c[ ], int na, int nb)
/* a[ ] and b[ ] are input sorted arrays */
/* c[ ] is the output sorted array after a[ ] and b[ ] are merged */
/* na and nb are the lengths of a[ ] and b[ ], respectively */
//You can call twoway(…) directly.

int merge_sort(….)    //complete the parameters by yourself
// merge_sort(….) is a recursive function.
{
Please write the body of the function.
} // end of merge_sort ( )
```