

CSD2181/2183 – Data Structure

Exercises

HUA Guang (华光)

Associate Professor

guang.hua@singaporetech.edu.sg

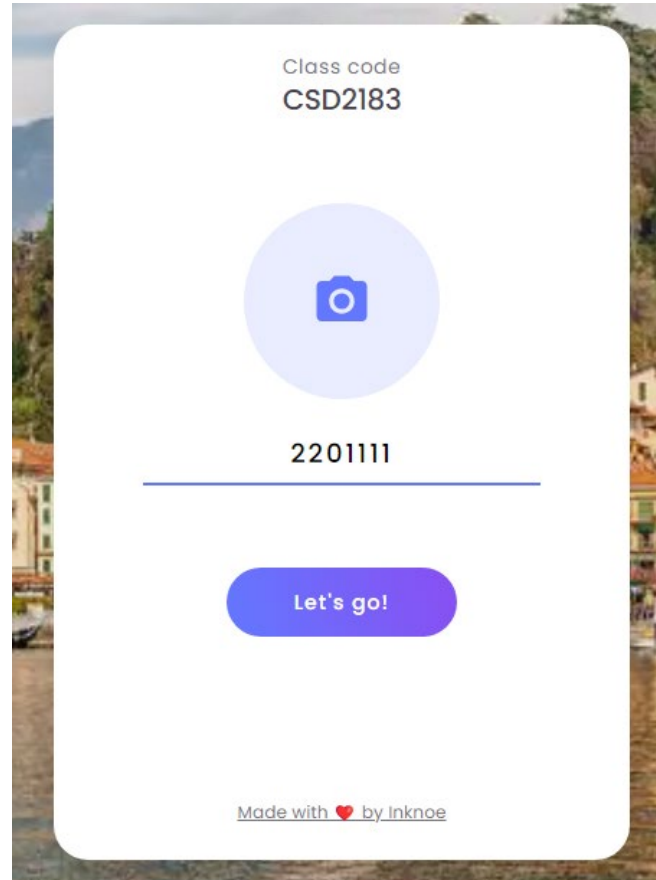
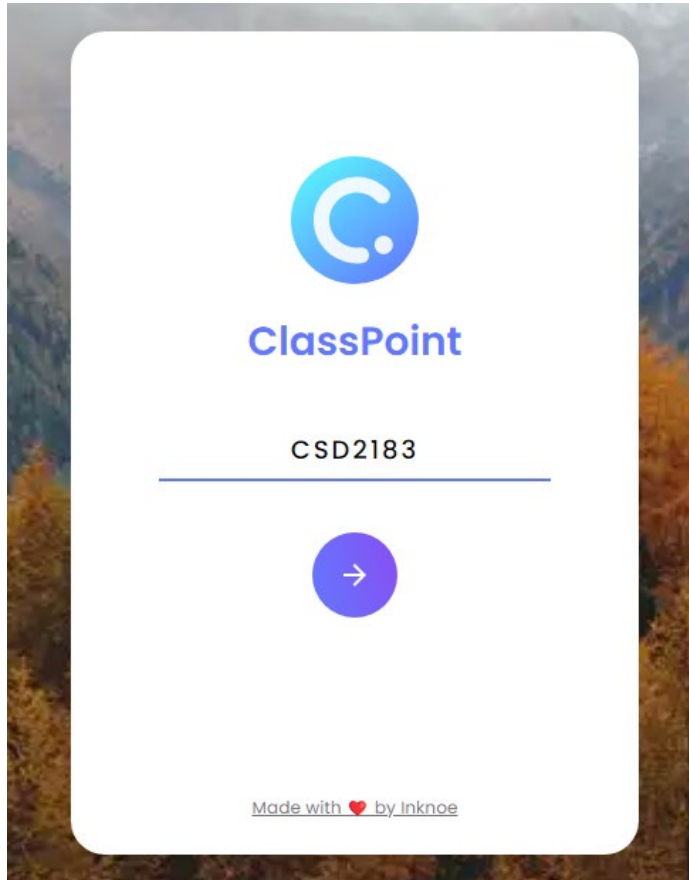


Introduction – Data Structure Exercises

- Purpose: to reinforce what you have learned and practiced in lectures.
- The exercise session is conducted face to face in class.
- It consists of a few MCQs to be solved within class.
- Limited time is given for each question (answer will be discussed afterwards).
- You are required to login to ClassPoint with your student ID.
- So, bring along your laptop or devices with Internet access.
- Attendance is compulsory and there is no make up.
- Exercises are marked considering your overall performance in the module.

Introduction – Data Structure Exercises

<https://www.classpoint.app/>



OR



Exercise 3

Sorting Algorithms

Exercise 3 – Sorting Algorithms

3.1 Which is the correct sequence?

- A. bedfae
- B. acebdf
- C. cbedfa
- D. bcedfa**

★ Multiple Choice

e	c	b	d	f	a
---	---	---	---	---	---

$i=1$

c	e	b	d	f	a
---	---	---	---	---	---

$i=2$

?	?	?	?	?	?
---	---	---	---	---	---

$i=3$

state of array
entering outer loop
iteration of
INSERTION SORT

What is the correct sequence in this iteration?

Exercise 3 – Sorting Algorithms

3.2 Which is the correct sequence?

- A. bacdfe
- B. acdbfe
- C. acbdef
- D. abcdfe**

★ Multiple Choice

e	c	b	d	f	a
---	---	---	---	---	---

$i = 0$

state of array
entering outer loop
iteration of

a	c	b	d	f	e
---	---	---	---	---	---

$i = 1$

SELECTION SORT

?	?	?	?	?	?
---	---	---	---	---	---

$i = 2$

What is the correct sequence in this iteration?

Exercise 3 – Sorting Algorithms

3.3 Which is the correct sequence?

- A. bacdfe
- B. bcdaef**
- C. cbdaef
- D. acbdef

★ Multiple Choice

e	c	b	d	f	a
---	---	---	---	---	---

$i = 0$

state of array
entering outer loop
iteration of

c	b	d	e	a	f
---	---	---	---	---	---

$i = 1$

BUBBLE SORT

?	?	?	?	?	?
---	---	---	---	---	---

$i = 2$

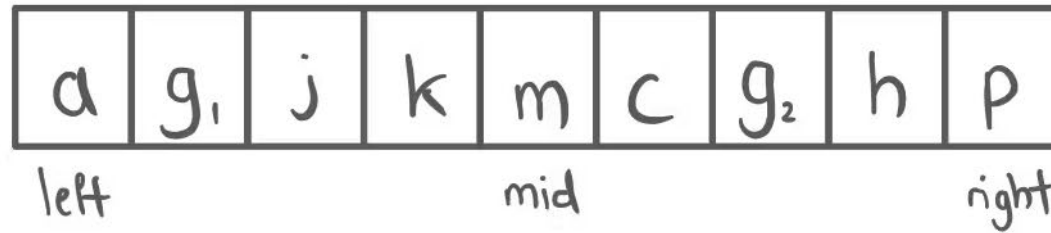
What is the correct sequence in this iteration?

Exercise 3 – Sorting Algorithms

3.4 Which is the correct sequence?

★ Multiple Choice

(A) cag2g1jhmkp; (B) acg1g2jkmhp; **(C) acg1g2hjkmp**; (D) acg2g1hjkmp



What is the resulting state of the array after performing `merge(left, mid, right)` on the above using MERGE SORT.

Exercise 3 – Sorting Algorithms

3.5 Which is the correct sequence?

 **Multiple Choice**

(A) adebc; **(B) abced**; (C) bacde; (D) abcde



What is the resulting state of the array after performing partition (start, end) on the above using QUICK SORT with $p[\text{end}]$ always chosen as the pval.

Exercise 3 – Sorting Algorithms

3.6 Which is the time complexity?

- A. $O(1)$
- B. $O(\log n)$
- C. $O(n)$**
- D. $O(n \log n)$
- E. $O(n^2)$
- F. $O(n^3)$

★ Multiple Choice

```
void facttail ( unsigned n , unsigned a ) {  
    if ( n == 0 )  
        return a ;  
    return facttail ( n - 1 , n * a ) ;  
}
```

What is the TIME complexity of facttail?

Exercise 3 – Sorting Algorithms

3.7 Which sorting algorithm builds the final sorted array one item at a time, taking each element from the list and inserting it into its correct position in the already-sorted part of the array?

- A. Bubble Sort
- B. Merge Sort
- C. Quick Sort
- D. Insertion Sort**



Multiple Choice

Exercise 3 – Sorting Algorithms

3.8 What type of sorting algorithm recursively divides the unsorted list into n sub-lists, each containing one element, and then repeatedly combine sub-lists to produce new sorted sub-lists?

- A. Bubble Sort
- B. Merge Sort**
- C. Quick Sort
- D. Insertion Sort



Multiple Choice

Exercise 3 – Sorting Algorithms

3.9 In which sorting algorithm does the list get divided into a sorted and an unsorted region, and the algorithm repeatedly selects the smallest (or largest) element from the unsorted region and swaps it with the first element of the unsorted region?

- A. Quick Sort
- B. Bubble Sort**
- C. Insertion Sort
- D. Selection Sort



Multiple Choice

Exercise 3 – Sorting Algorithms

3.10 Which of the following sorting algorithms feature the divide-and-conquer strategy?

- A. Merge Sort Only
- B. Bubble Sort Only
- C. Quick Sort Only
- D. Quick Sort and Insertion Sort
- E. Quick Sort and Merge Sort**

 Multiple Choice

Exercise 3 – Sorting Algorithms

3.11 Which of the following sorting algorithms repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order until the list is sorted?

- A. Selection Sort
- B. Merge Sort
- C. Bubble Sort**
- D. Insertion Sort



Multiple Choice

Exercise 3 – Sorting Algorithms

Summary Example 1: Sort {36, 28, 31, 26, 61, 37} ASC

Bubble Sort

36, 28, 31, 26, 61, 37
 28, 36, 31, 26, 61, 37
 28, 31, 36, 26, 61, 37
 28, 31, 26, 36, 61, 37
 28, 31, 26, 36, 37, 61
 28, 26, 31, 36, 37, 61
 26, 28, 31, 36, 37, 61
 26, 28, 31, 36, 37, 61

Input/Output

Unsorted

Swapped

Selection Sort

36, 28, 31, 26, 61, 37
 26 | 28, 31, 36, 61, 37
 26, 28 | 31, 36, 61, 37
 26, 28, 31 | 36, 61, 37
 26, 28, 31, 36 | 61, 37
 26, 28, 31, 36, 37 | 61
 26, 28, 31, 36, 37, 61

Input/Output

Sorted

Unsorted

Swapped

Insertion Sort

36, 28, 31, 26, 61, 37
 28, 36, 31, 26, 61, 37
 28, 31, 36, 26, 61, 37
 28, 31, 26, 36, 61, 37
 28, 26, 31, 36, 61, 37
 26, 28, 31, 36, 61, 37
 26, 28, 31, 36, 61, 37
 26, 28, 31, 36, 37, 61
 26, 28, 31, 36, 37, 61

Input/Output

Sorted

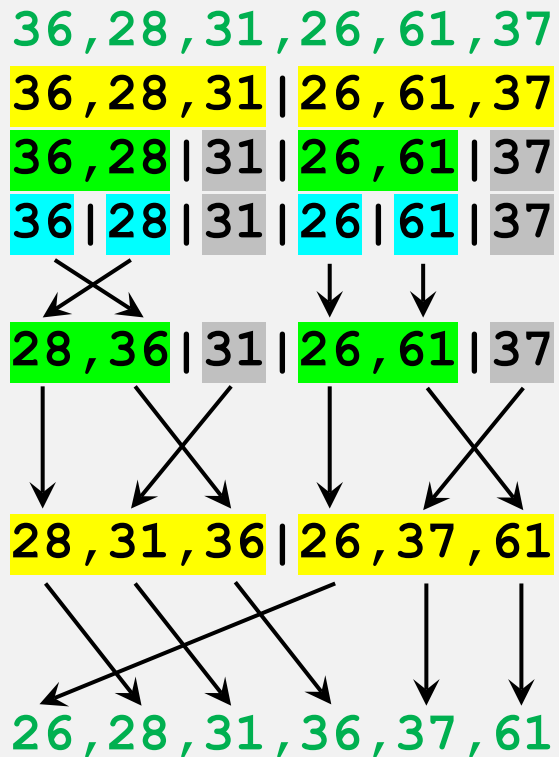
Unsorted

Swapped

Exercise 3 – Sorting Algorithms

Summary Example 1: Sort {36, 28, 31, 26, 61, 37} ASC

Merge Sort



Quick Sort

(Pivot = First) →

36, 28, 31, 26, 61, 37
 36, 28, 31, 26, 61, 37
 36, 28, 31, 26, 61, 37
 36, 28, 31, 26, 61, 37
 36, 28, 31, 26, 61, 37
 36, 28, 31, 26, 61, 37
 36, 28, 31, 26, 61, 37
 36, 28, 31, 26, 61, 37
 26, 28, 31 | 36 | 61, 37
 26, 28, 31 | 36 | 61, 37
 26 | 28, 31 | 36 | 37 | 61
 26 | 28 | 31 | 36 | 37 | 61
 26, 28, 31, 36, 37, 61

Unsorted, Pivot
 Smaller, Bigger
 Input/Output/Sorted

Quick Sort

(Pivot = Last) →

36, 28, 31, 26, 61, 37
 36, 28, 31, 26, 61, 37
 36, 28, 31, 26, 61, 37
 36, 28, 31, 26, 61, 37
 36, 28, 31, 26, 61, 37
 36, 28, 31, 26, 61, 37
 36, 28, 31, 26, 61, 37
 36, 28, 31, 26, 61, 37
 36, 28, 31, 26 | 37 | 61
 36, 28, 31, 26 | 37 | 61
 26 | 28, 31, 36 | 37 | 61
 26 | 28, 31, 36 | 37 | 61
 26 | 28, 31, 36 | 37 | 61
 26 | 28, 31, 36 | 37 | 61
 26 | 28, 31, 36 | 37 | 61
 26, 28, 31, 36, 37, 61

Exercise 3 – Sorting Algorithms

Summary Example 2: Sort {36, 31, 28, 61, 26, 37} ASC

Bubble Sort

36, 31, 28, 61, 26, 37
 31, 36, 28, 61, 26, 37
 31, 28, 36, 61, 26, 37
 31, 28, 36, 26, 61, 37
 31, 28, 36, 26, 37, 61
 28, 31, 36, 26, 37, 61
 28, 31, 26, 36, 37, 61
 28, 26, 31, 36, 37, 61
 26, 28, 31, 36, 37, 61
 26, 28, 31, 36, 37, 61

Input/Output
 Unsorted
 Swapped

Selection Sort

36, 31, 28, 61, 26, 37
 26 | 31, 28, 61, 36, 37
 26, 28 | 31, 61, 36, 37
 26, 28, 31 | 61, 36, 37
 26, 28, 31, 36 | 61, 37
 26, 28, 31, 36, 37 | 61
 26, 28, 31, 36, 37, 61

Input/Output
 Sorted
 Unsorted
 Swapped

Insertion Sort

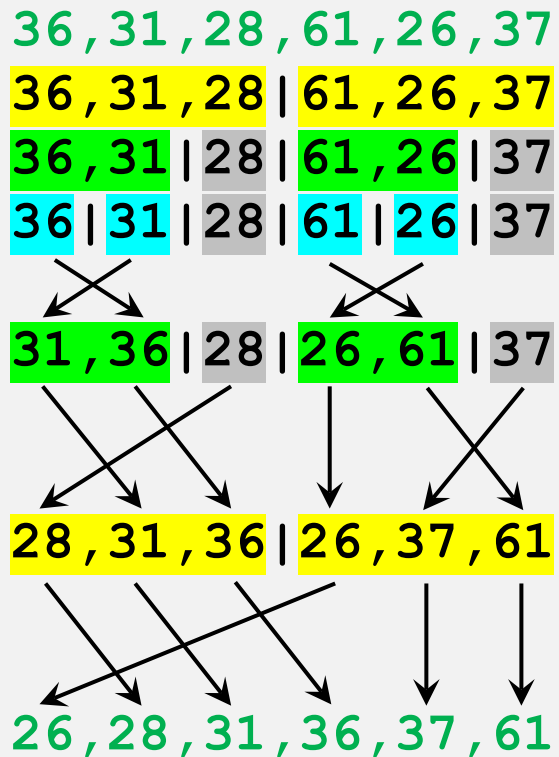
36, 31, 28, 61, 26, 37
 31, 36, 28, 61, 26, 37
 31, 28, 36, 26, 61, 37
 28, 31, 36, 26, 61, 37
 28, 31, 26, 36, 61, 37
 28, 26, 31, 36, 61, 37
 26, 28, 31, 36, 61, 37
 26, 28, 31, 36, 37, 61
 26, 28, 31, 36, 37, 61

Input/Output
 Sorted
 Unsorted
 Swapped

Exercise 3 – Sorting Algorithms

Summary Example 2: Sort {36, 31, 28, 61, 26, 37} ASC

Merge Sort



Quick Sort

(Pivot = First) →

```

36, 31, 28, 61, 26, 37
36, 31, 28, 61, 26, 37
36, 31, 28, 61, 26, 37
36, 31, 28, 61, 26, 37
36, 31, 28, 61, 26, 37
36, 31, 28, 26, 61, 37
36, 31, 28, 26, 61, 37
26, 31, 28 | 36 | 61, 37
26, 31, 28 | 36 | 61, 37
26 | 31, 28 | 36 | 37 | 61
26 | 28 | 31 | 36 | 37 | 61
26, 28, 31, 36, 37, 61
  
```

Unsorted, **Pivot**
Smaller, **Bigger**
 Input/Output/Sorted

Quick Sort

(Pivot = Last) →

```

36, 31, 28, 61, 26, 37
36, 31, 28, 61, 26, 37
36, 31, 28, 61, 26, 37
36, 31, 28, 61, 26, 37
36, 31, 28, 61, 26, 37
36, 31, 28, 61, 26, 37
36, 31, 28, 26, 61, 37
36, 31, 28, 26, 61, 37
36, 31, 28, 26 | 37 | 61
36, 31, 28, 26 | 37 | 61
26 | 31, 28, 36 | 37 | 61
26 | 31, 28 | 36 | 37 | 61
26 | 28 | 31 | 36 | 37 | 61
26, 28, 31, 36, 37, 61
  
```

The End