# Beijing-Dublin International College

---

**SEMESTER I FINAL EXAMINATION - 2017/2018**

---

**School of Computer Science**

**COMP2002J Data Structures and Algorithms 1**

HEAD OF SCHOOL: Pádraig Cunningham
MODULE COORDINATOR: Lina Xu*

**Time Allowed: 120 minutes**

**Instructions for Candidates**

All questions carry equal marks. The distribution of marks in the right margin shown as a percentage gives an approximate indication of the relative importance of each part of the question.

**BJUT Student ID: _____** **UCD Student ID: _____**

I have read and clearly understand the Examination Rules of both Beijing University of Technology and University College Dublin. I am aware of the Punishment for Violating the Rules of Beijing University of Technology and/or University College Dublin. I hereby promise to abide by the relevant rules and regulations by not giving or receiving any help during the exam. If caught violating the rules, I accept the punishment thereof.

**Honesty Pledge：_____ (Signature)**

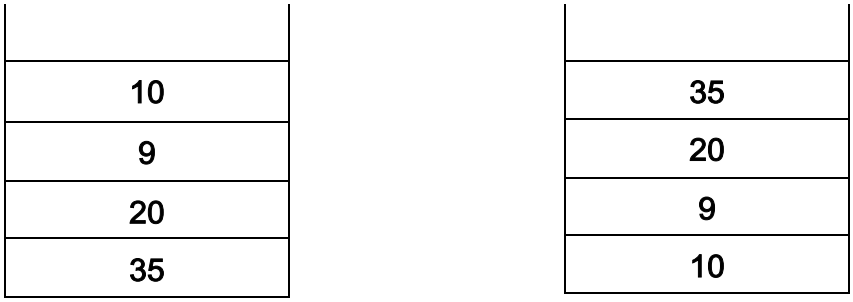
**Instructions for Invigilators**

Non-programmable calculators are permitted.
No rough-work paper is to be provided for candidates.

| Obtained score |
| --- |
| 10 |

## Question 1: Stack

a. What is the sequence should be if you want to get the stack structure A through push operation? How you can operate on the stack A to get stack B? Write your answer in Java or Pseudocode. (8 Marks)




b. Array based implementation (size=10), what the array should be like after executing the following code? (2 Marks)

push(5); push(4); pop(); top(); push(1); push(8); push(7); push(2); pop();

| | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | | | | | | | | | |

| Obtained score |
| --- |
| 20 |

## Question 2: List

a. We have studied 3 implementations of the List abstract data type: single linked list, array-based list and double linked list. Among these three, which one is the most efficient one? Explain your answer by comparing the complexity of some operations in the three implementations. (5 Marks)

b. Supposedly *fileFolder* is an instance from Double-Linked List. Now you have ten *files* (instanced from *MyFile* class) that are required to insert into *fileFolder.* This is how you create a *file* from *MyFile* class:

MyFile file = new MyFile(1, “Blue”).

The first parameter is the unique file number and the second parameter indicates the file's colour. There are only three colours among the ten files: blue, red and black. After insertion the ten files, all the same colour should be grouped together, while the order of the colours does not matter. The ten files are:

(1, "Blue"); (2, "Red"); (3, "Blue"); (4, "Black"); (5, "Blue");
(6, "Black"); (7, "Red"); (8, "Blue"); (9, "Red"); (10, "Red").

The order for inserting the files should be from 1 to 10. How can you achieve the goal ONLY using the functions provided in DLL? Write your answer in Java or pseudocode.

(10 Marks)

c. In array based implementation for List. We use *ArrPos* class implemented *Position* Interface. What values are stored in *ArrPos* and why do we need them?

(5 Marks)

| Obtained score |
| --- |
| 20 |

## Question 3: Queue

a. What is the complexity to find the maximal value in a Queue with all unsorted integer values? (2 Marks)

b. What is the fundamental difference between Queue and List? (3 Marks)

c. Suppose that a client performs an intermixed sequence of insertion and removing for a Double Ended Queue. The queue originally has a storage of number from 0-9. When removing from the queue, the number will be printed. Which of the following output sequences is (are) impossible? (5 Marks)

- 0 1 2 3 4 5 6 7 8 9
- 0 9 8 1 7 6 5 2 3 4
- 4 6 8 7 5 3 2 9 0 1

d. In a bank, the customers will join a queue to get services. Supposedly in this branch, there are two windows open for serving customers (which can be interpreted as two queues). When a VIP customer comes to the bank, he/she can join the queue which has the smaller size from the front without waiting. Implement a class called *Bank* to simulate this scenario in this bank. (10 Marks)

(Hints: this bank should maintain two queues. There should be functions to support operations on both queues. An additional function *VIPJoin* should be implemented.)

| Obtained score |
| --- |
| 15 |

## Question 4: Map

a. What is the complexity for the following code? (5 Marks)

```
1. Map<Integer, Integer> map =
2. new HashMap<Integer, Integer>();
3. for (int i = 1; i <= N; i++) {
4.     map.put(i, i+1);
   }
5. int sum = 0;
6. for (int i : map.keySet()) {
7. Set<Integer> copy =
8. new HashSet<Integer>(map.values());
9. if (copy.contains(i * 2)) {
10.          sum++;
          }
      }
11.  System.out.println(sum);
```

b. There are two main types of collision handling strategies. What are they and how they work? (10 Marks)

| Obtained score |
| --- |
| 35 |

## Question 5: Sorting and Complexity

a. What are the complexities of the traditional sorting methods: selection sort, rank sort and insertion sort in big O notation? What are the complexities of the advanced sorting methods: quick sort and merge sort in big O notation? (5 Marks)

b. Trace the execution of the quick sort algorithm over the array below, using the first element as the pivot. Show each pass of the algorithm, with the pivot selection and partitioning, and the state of the array as/after the partition is performed, until the array is sorted. (Every time there is a change on the array) (10 Marks)

| | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 16 | 21 | 8 | 11 | 53 | 3 | 26 | 49 | 31 |

c. I am sorting data that is stored over a network connection. Based on the properties of that connection, it is extremely expensive to "swap" two elements. But looping over the elements and looking at their values is very inexpensive. I want to minimize swaps above all other factors. Which sorting algorithm(s) can you use? At least write down one. (5 Marks)

d. I have an array that is already sorted. Periodically, some new data comes in and is added to the array at random indexes, messing up the ordering. I need to re-sort the array to get it back to being fully ordered. I do not want to use very much additional memory during the sort. Which sorting algorithm(s) can you use? At least write down one. (5 Marks)

e. How can you use Singly-Linked list to implement Radix Sort? You can write words, diagrams, pseudocode or Java to describe your idea. (15 Marks)