Beijing-Dublin International College

SEMESTER I RESIT 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)**

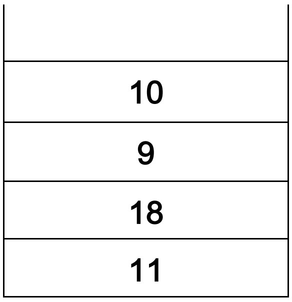
Instructions for Invigilators

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

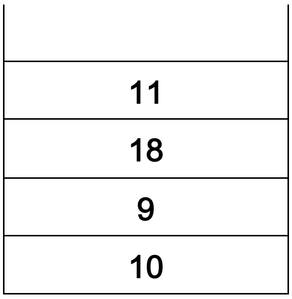
Obtained score
10

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 if you do not know what numbers are actually stored in A? Write your answer in Java or Pseudocode. (8 Marks)



A



B

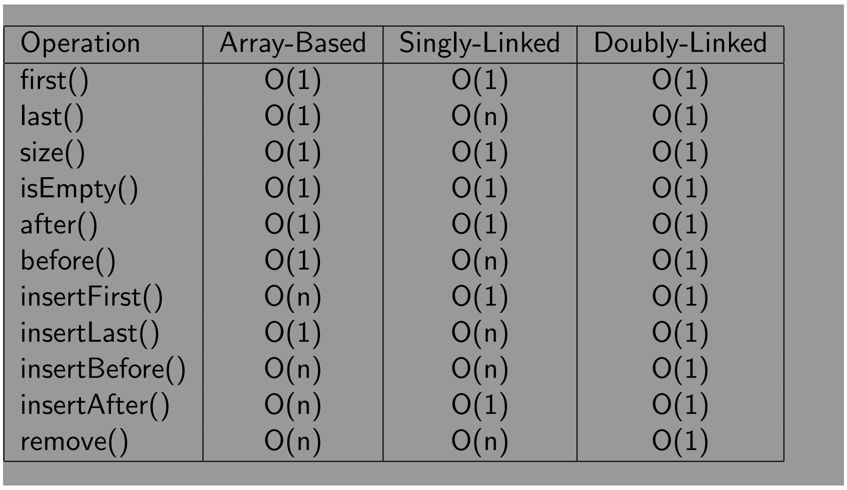
b. Array based implementation (size=10), what the array should be like after executing the following code? (2 Marks)

push(5); push(4); push(9); pop(); pop(); top(); push(1); push(8); 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)



Operation	Array-Based	Singly-Linked	Doubly-Linked
first()	O(1)	O(1)	O(1)
last()	O(1)	O(n)	O(1)
size()	O(1)	O(1)	O(1)
isEmpty()	O(1)	O(1)	O(1)
after()	O(1)	O(1)	O(1)
before()	O(1)	O(n)	O(1)
insertFirst()	O(n)	O(1)	O(1)
insertLast()	O(1)	O(n)	O(1)
insertBefore()	O(n)	O(n)	O(1)
insertAfter()	O(n)	O(1)	O(1)
remove()	O(n)	O(n)	O(1)

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.

(15 Marks)

Obtained score
20

Question 3: Queue

a. What is the fundamental difference between Queue and List? (5 Marks)

b. Suppose that a client performs a sequence of insertion from 0-9 to a Double Ended Queue. Which of the following result is impossible? (5 Marks)

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

c. 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 come to the back, 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 operation 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)

```
for ( i = 0; i < N; i++ ) {
  for ( j = 0; j < N; j++ )
   System..out.println( “Big O” ) ;
}
```

b. What is open addressing and what is used for? Explain in your words or diagrams. (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? (10 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

Answer: pivot =16

16	31	8	11	53	3	26	49	21
16	49	8	11	53	3	26	32	21

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

c. 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)