National University of Singapore

School of Computing

CS2040C - Data Structures and Algorithms

Midterm Quiz @ I3-AUD

(Thu, 19 Sep 2019, S1 AY2019/20, 40m)

INSTRUCTIONS TO CANDIDATES:

1. Do **NOT** open this assessment paper until you are told to do so.
2. This assessment paper contains FIVE (5) sections.
   It comprises SIX (6) printed pages, including this page.
3. This is an **Open Book Assessment**.
   Any electronic device **is not allowed**.
4. Answer **ALL** questions within the **boxed space** in this booklet.
   You can use either pen or pencil. Just make sure that you write **legibly**!
5. Write your Student Number in the box below:

A	0							

Section	Maximum Marks	Student Marks	Remarks
A	20		
B	20		
C	25		
D	25		
E	10		
Total	100		

A Worst Case Time Complexity Analysis (20 marks)

Write down the *tightest*[1] *worst case* time complexity of the various data structure operations or algorithms below. Each correct answer worth 5 marks.

The operations (algorithms) referred below are the **unmodified version**, as per discussion in class, e.g. as currently explained in VisuAlgo or as currently implemented in C++ STL. Unless otherwise mentioned, there are currently n (not necessarily distinct) integers ($100\,000 \leq n \leq 200\,000$) in the data structure: a `std::vector v`.

No	Operations		Time Complexities
1	Insert a new integer x at the *middle* position of an unsorted `v` (retain the order in `v`)		$O(\_\_\_\_\_\_\_)$
2	Search the location of a value `x` in a **sorted** `v`		$O(\_\_\_\_\_\_\_)$
3	Run `std::stable_sort` on a **sorted** `v`		$O(\_\_\_\_\_\_\_)$
4	Find **three** different integers that sum to a target value `x` in **sorted** `v`		$O(\_\_\_\_\_\_\_)$

[1] What we meant by tightest worst case time complexity is as follows: If an operation of the stipulated data structure/an algorithm needs at best $O(n^3)$ if given the worst possible input but you answer higher time complexities than that, e.g. $O(n^4)$ – which technically also upperbounds $O(n^3)$, you will get wrong answer for this question.

Section A Marks = _____

B Sorting Review (20 marks)

There are five short statements below. The answers have been discussed in live class. Each correct statement worth 4 marks. **No answer** is a possible answer.

1. After learning several sorting algorithms so far and after attempting PE1 yesterday, I declare that the best sorting algorithm is (write the algorithm name): ____________________ because (write some explanation): ____ ____________________________________.

2. Mention the name of any $O(n)$ **comparison-based** sorting algorithm: _______________ (assume that n is relatively big and cannot be treated as constant factor). Follow-up question: Mention the name of any $O(n)$ sorting algorithm: _______________.

3. Mention a simple way to make the default implementation of Merge Sort that has been shown in class/VisuAlgo–which is stable–to be non stable instead: ______________________ ____.

4. Show how to make any non-stable sorting algorithm, e.g. Randomized Quick Sort, to enforce stable output! Hint: You are allowed to modify the input (especially involving the duplicates inside the input): ____ ____.

5. What are the implications if we change the recursive part of Quick Sort to recursively sort the right subarray first before we recursively sort the left subarray like the one shown in `quickSort2` routine below? ____________________________________ (assume that the partition routine is the correct randomized implementation, including if there are many duplicates).

```
void quickSort2(int a[], int low, int high) {
  if (low < high) {
    int m = partition(a, low, high); // O(N)
    // a[low..high] ~> a[low..m-1], pivot, a[m+1..high]
    quickSort2(a, m+1, high); // we recursively sort right subarray first
    // a[m] = pivot is already sorted after partition
    quickSort2(a, low, m-1); // then recursively sort left subarray next...
  }
}
```

C A 'New' ADT (25 marks)

Suppose you are tasked to design a 'New' Abstract Data Type (ADT) that needs to support the following four crucial operations.

1. `add(v)` - add item/data `v` into your ADT (you can assume that `v` is a non-negative integer),
2. `count()` - return the number of items currently in your ADT,
3. `returnAny()` - return *any* item (any integer) that is currently in your ADT, or -1 (to signal that there is nothing currently in your ADT),
4. `removeAny()` - remove *any* item (any integer) that is currently in your ADT.

Please select your *best* data structure to implement this 'New' ADT and its four crucial operations.
You can use any data structure that has been discussed in CS2040C (or beyond).
Please describe how you will actually implement these four crucial operations.
Please also analyze their time complexities!
You will be graded based on the efficiency of your implementation (25 marks for the best answers).

D Emulating a Queue with Two Stacks (25 marks)

You probably have not heard about it, but it is actually possible to emulate a Queue FIFO (First-In First-Out) behavior using only **two** Stacks, yet maintain $O(1)$ enqueue and dequeue performance. Recall that a Stack is a LIFO (Last-In First-Out) data structure, which is the opposite of what we want to do with a Queue (FIFO).

(15 marks): The question is, how to do this? You can use Figure 1 to help you figure out the way.

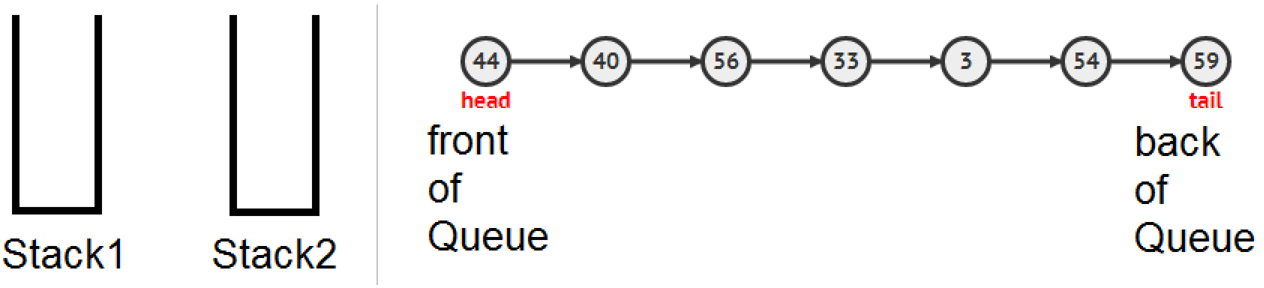

Figure 1: Illustration

Follow up question 1 (5 marks): Let's assume that we already know how to emulate a Queue with two Stacks. Show how to emulate a Queue with two **Vectors (resizeable arrays)** instead of two Stacks?

Follow up question 2 (5 marks): So is it still important to study Linked List data structure knowing that both Stack and Queue ADT can be efficiently implemented using **Vectors (resizeable arrays)**?

E Easy Marks (10 marks)

To qualify for up to easy 10 marks, you need to **select all three full names correctly**.
The name of my CS2040C lecturer is ______________________________,
The name of my CS2040C **tutorial** Teaching Assistant is ______________________________,
The name of my CS2040C **laboratory** Teaching Assistant is ______________________________.
The choices for these MCQs: Joshua Casey Darian, Kuan Wei Heng, Matthew Ng Zhen Rui, Modak Shantanu Bharat, Srivastava Aaryam, Steven Halim, Steven Wijaya, or Quah You Jing Kane.

Write a **short** (maybe limit yourself to up to 3 minutes to do this and about 3-4 sentences) **but honest (and not anonymous)** feedback on what you have experienced in the first 6 weeks of CS2040C in Semester 1 AY 2019/20 (including Week -02/-01 experience, if any). Feedback that are shared by *majority* (**not a one-off**) and can be easily incorporated to make the next 7 weeks of CS2040C better will be done. Grading scheme: 0-blank, 4/5-considered trivial feedback but not blank, 9/10-good and constructive feedback, thanks. (Penalty -2 marks for each wrongly selected name above).

– End of this Paper, All the Best –