University of Toronto Mississauga

Midterm Test

Course: CSC263H5 Winter 2016

Instructor: Larry Zhang

Duration: 50 minutes

Aids allowed: 1 double-sided 8.5" x 11" aid sheet

Last Name: _____

Given Name: _____

Flip to the back cover and write down your name and student number.

This midterm consists of a total of 44 marks, 3 rounds on 10 pages (including this one). When you receive the signal to start, please make sure that your copy is complete.

Each question is labelled with the suggested amount of time that you should spend on it. You may use it as reference to better manage your time.

Precise answers will be given higher marks than vague ones. Concise answers will be given higher marks than lengthy ones. Illegible answers will not be given marks.

If you write any answer on the pages for rough works, indicate clearly what you want marked. If you need extra scratch paper, raise your hand, and the instructor will bring you a few sheets.

Trust yourself, you know more than you think you do!

blank page for rough works which will not be marked

THE JOB INTERVIEW

We said that one goal of this course is to learn to be able to nail job interviews. So today you will go through several rounds of interviews by the giant tech company -- Hooli, whose corporate motto is: "No one shall make the world a better place better than we do". (Source: Silicon Valley, HBO)

The good news: If you answer the questions well enough, you will get the job.

The bad news: This is not true. It is just a midterm.

Good luck!

Round 1: Phone Interview with Human Resource [2x8=16 marks] [8 minutes]
The answers to these questions are so short that even a HR person can handle them.

1. True or False? To prove that the best-case runtime is upper-bounded by T(n), we just need to find one input that has runtime smaller than T(n).

(a.) true b. false c. neither of above

2. Which of the following functions grows the **fastest**? Circle your answer.

a. $log\ n$ b. $log\ log\ n$ (c. $log^2\ n$) d. $\sqrt{log\ n}$

3. How many **leaf nodes** are there in a binary heap with 263 nodes? Write down your answer in the space below.

132

4. Which of the following data structures achieves the best performance for implementing a priority queue on which we will **only** do INSERT? Circle your answer.

a. sorted array b. sorted linked list (c. unsorted linked list) d. heap

5. True or False? In a BST, the predecessor of a node x must be in the left subtree of x.

a. true (b. false) c. neither of above

6. Suppose we have an algorithm with worst-case runtime $O(n^3)$ and average-case runtime O(n), what is the technique that we can use to guarantee that we get the O(n) runtime most of the time in practice?

Randomization

7. Compared to a hash table, using a direct-address table could cost much more ...

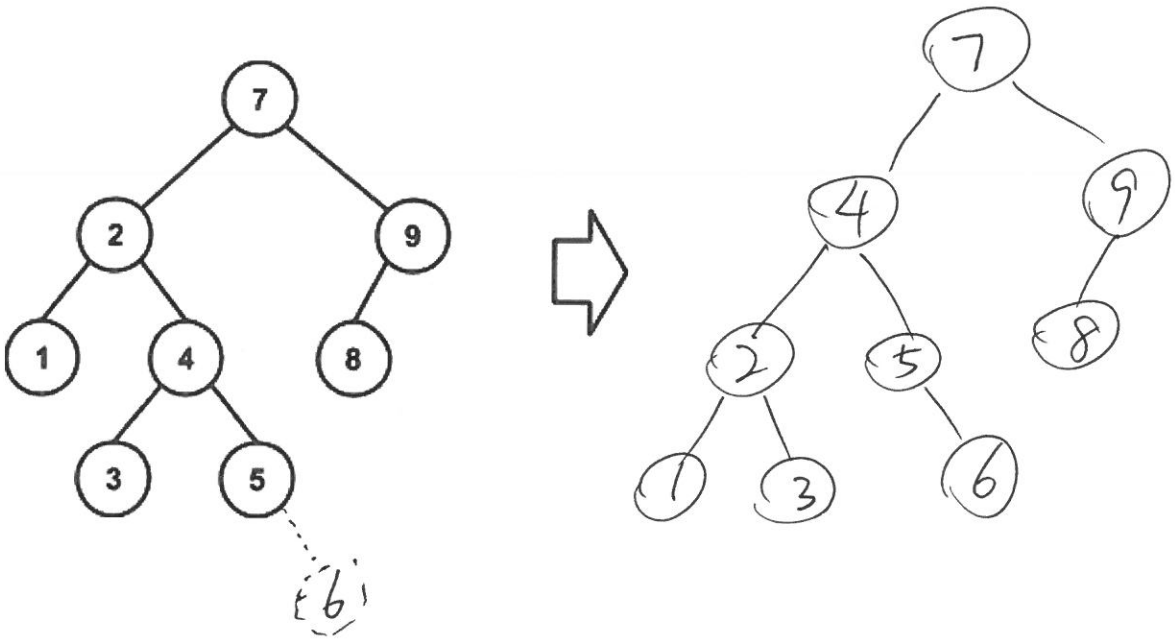
(a.) space b. time c. both d. neither

8. In the lecture we learned that, if we expand the dynamic array by **doubling** the size when it is full, the amortised cost per operation of a sequence of APPEND operations is 3. If we change the expansion rule to **tripling** (three times) the size when it is full. What is the amortised cost per operation? Write down your answer in the space below.

2.5

Round 2: On-Site Interview with Software Engineers [22 marks] [25 minutes]
The answers are supposed to be short. Try to make your answers as concise as possible.

1. Given the following AVL tree, **draw** the resulting AVL tree after **inserting a new node with key 6**. [4 marks]



2. Consider an open addressing hash table with **m** buckets, and consider the following sequence of n keys: **1, 3, 5, 7, ..., 2n-1** (assume n < m). **Specify** a hash function **h(k)** and a type of probing, so that inserting the **n-th** key (2n-1) needs to visit **n** buckets in the hash table. [4 marks]

$h(k) = 1$

linear probing

3. Given two binary max-heaps stored in arrays H1 and H2. How do you **merge** the two heaps into one new binary max-heap H3 (i.e., H3 has the combination of elements in H1 and H2, and is heap-ordered), under runtime O(H1.size + H2.size) in the worst case? Step 1 is given to you below, you just need to complete Step 2. [4 marks]

Step 1: Create array H3 whose content is the concatenation of H1 and H2, i.e., the content of H3 is H1's content followed by H2's content.

Step 2: Call BuildMaxHeap(H3)

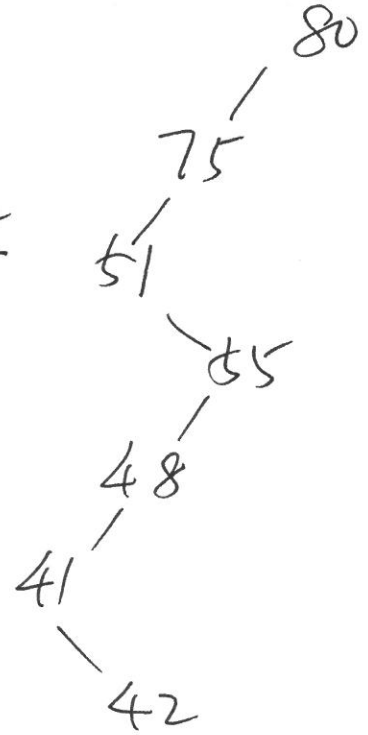
4. We want to check whether a BST is broken ("broken", as in the problem set, means not all nodes in the tree satisfy the BST property). The tool we have is that we can perform a SEARCH operation on the BST and get the sequence of nodes that are visited during the search. So we performed **SEARCH(42)** and got the following sequence:

80 (root), 75, 51, 55, 48, 41, 42 (found)

Can you tell for sure whether the BST is broken from the above sequence? Justify your answer. [4 marks]

Yes, broken

because 48 is in the right subtree of ~~51~~ 51



5. Considering the follow algorithm which searches for the last appearance of value 42 in an array A of length n (assume n > 42). The index of the array starts from 1.

```
FindLast(A):
1     for i = n, n-1,... downto 1:
2         if A[i] == 42:    # count this line only
3             return i
4     return 0
```

The input array A is generated in a specific way: A[1] is always 1, A[2] is picked from the set {1, 2} uniformly at random; A[3] is picked from the set {1, 2, 3} uniformly at random;..., A[n] is picked from the set {1, 2, ..., n} uniformly at random. All picks are made independently.

Now we want to analyse the runtime of `FindLast()` by counting the number of comparisons made, i.e., the number of times Line #3 is executed. Answer the following questions. Justification is NOT required. [6 marks]

(1) What is the best-case runtime? [1 mark]

1

(2) What is the probability of the best case happening? [2 marks]

$\frac{1}{n}$

(3) What is the worst-case runtime? [1 mark]

n

(4) What is the probability of the worst-case happening? (Note that we assume n > 42) [2 marks]

$\frac{41}{n}$

Round 3: Final Interview with the CTO -- Bob. [6 marks] [8 minutes]
You will get 1 out of 6 marks for leaving this question completely blank.

Let $a_1, a_2, ..., a_n$ be a sequence of real numbers, for $n \geq 1$. A **SumSet** is an ADT which stores the sequence and supports the following operations:

- → PARTIAL-SUM(S, m): return $\sum_{i=1}^{m} a_i$, the partial sum from a_1 to $a_m (1 \leq m \leq n)$.
- → CHANGE(S, i, y): change the value of a_i to a real number y.

Design a data structure that implements SumSet, using Bob's favourite data structure: **augmented AVL tree**. The worst-case running times of both PARTIAL-SUM and CHANGE must be in O(logn). Describe your design by answering the following questions.

(a) What is the sorting **key** of each node of the AVL tree? [2 mark]

The sequence index i

(b) What is the **additional attribute** at each node which makes it possible to compute PARTIAL-SUM in O(log n) time? Briefly justify why this attribute can be maintained efficiently upon changes to the tree. [2 marks]

X.Sum
Sum of node's values (ai) in the
subtree rooted at X
X.Sum only depends on X.left.Sum, X.right.Sum and X.value
So it is maintainable within O(logn) time.

(c) The **pseudocode** of the PARTIAL-SUM should be very similar to an operation that we learned in this course. Which operation is it? Just write down its name. [2 marks]

RANK

blank page for rough works which will not be marked

Last Name: _____

Given Name: _____

Student Number: _____

R1: _____ / 16

R2: _____ / 22

R3: _____ / 6

TOTAL: _____ / 44

END OF TEST