CS 2150 Exam 2, fall 2021

Name

You MUST *clearly* write your name above. You must also write your e-mail ID on **EACH** page.

If you are still writing when "pens down" is called, your exam will not be graded – even if you are still writing to fill your name and userid. So please do that first. Sorry to have to be strict on this!

There are 6 pages to this exam. Once the exam starts, please make sure you have all the pages. Questions are worth different amounts of points.

Answers for the short-answer questions should not exceed about 20 words; if your answer is too long (say, more than about 30 words), you will get a zero for that question! This is not a hard maximum, but you should try your best to answer the question as concisely as you can.

This exam is CLOSED text book, closed-notes, closed-calculator, closed-cell phone, closed-computer, closed-neighbor, etc. Questions are worth different amounts, so be sure to look over all the questions and plan your time accordingly. Please sign the honor pledge below.

A crash reduces
Your expensive computer
To a simple stone.

Page 2: Exam 1 material

1. [6 points] Convert $-4.4375 = -4\frac{7}{16}$ into bit-Endian 32-bit IEEE 754 floating point notation. Show your work!

2. [6 points] Fill in the **worst-case** running times for the three primary operations for the listed data structures. Note that they should be interpreted appropriately for each data structure, so insert in a stack is really push, find on a queue is front, etc. Write the running time as just n, not $\Theta(n)$ and not "linear" (and appropriately differently if it's not linear). Any removal assumes there is *already* an iterator (or similar) to the element to be removed, if appropriate to that data structure. Any insert assumes you are inserting to the end of the list that is most efficient. *find()* is finding if an element is contained in the data structure, it is not find at index.

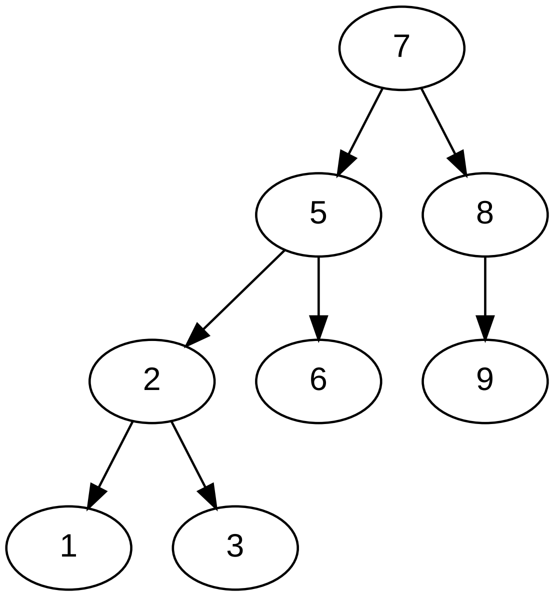
Data structure	find time	insert time	remove time
Singly linked list			
Doubly linked list			
Array-based list			
Vector-based list			
Stack, vector-based			
Queue, vector-based			
Stack, linked-list based			
Queue, linked-list based			

Page 3: Trees

3. [3 points] Why is an AVL tree generally slower than a red-black tree? Why doesn't the big-Theta analysis show this?

4. [3 points] When are splay trees a good data structure to use?

5. [6 points] Consider the following AVL tree. Draw the resulting tree when we insert 4.



Page 4: Hashes

6. [3 points] Write the pseudo-code for a hash function that violates *all three* of the required properties of a hash function. *Note: the hash function will likely be ridiculous. It should be!*

7. [3 points] Consider a hash table which uses linear probing and rehashes whenever the load factor is larger than .5. What are the big-theta **worst-case** running times for find and insert? *Make sure to consider the worst-case, including the cost to rehash if necessary.*

8. [6 points] Consider three hash tables of size 10. The keys are non-negative integers, and the hash function is $hash(k) = k\%10$. If needed, the secondary hash function is $hash_2(k) = ((k + 6)\%10) + 1$. Each table uses a different collision resolution strategy. Insert the elements listed below in into each of those hash tables, following the collision resolution strategy listed at the top of the column. The keys to insert are: 87, 21, 38, 53, and 27.

index	Linear probing	Quadratic probing	Double hashing
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			

Page 5: Assembly & IBCM

9. [3 points] In call-by-value, the copies of the parameters are discarded after the subroutine call. How are they discarded? *Make sure to incorporate the calling convention in your answer here.*

10. [3 points] What are the limitations from writing any algorithm we can think of in IBCM?

11. [6 points] Consider the following x86 function. First, state what is suppose to do (in 5 words or less!), and then fill in the missing three blanks.

```
mystry:
  xor r11, r11
  xor rax, rax
body:
  xor r10, r10
  cmp r11, rsi

  je _____________ (a)
  mov ecx, [rdi+4*r11]
  cmp edx, ecx
  je end

  inc _____________ (b)
  jmp body
earlyexit:
  mov rax, -1
  ret
end:
  mov _____________, r11 (c)
  ret
```

Page 6: Miscellaneous

12. [3 points] Give a real-world example of when you would want to use the SHA-256 hashing algorithm.

13. [3 points] What does *amortized* mean with regard to big-Theta analysis of a running time?

14. [3 points] Given a data set of 40 elements, and an load factor of 0.5, what is the ideal hash table size? *Assume you are using double hashing, and want a table size that prevents thrashing from occurring while still keeping the load factor at or below 0.5*

15. [3 points] List one thing that bash shell scripts are good for, and two things that they are not good for