CS 2150 Exam 2

Name _____

You MUST write your e-mail ID on **EACH** page and bubble in your userid at the bottom of this first page. And put your name on the top of this page, too.

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

Other than bubbling in your userid at the bottom of this page, please do not write in the footer section of this page.

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.

If you do not bubble in this first page properly, you will not receive credit for the exam!

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.

_____

_____

_____

_____

_____

The Tao that is seen
Is not the true Tao,
until You bring fresh toner.

(the bubble footer is automatically inserted into this space)

Page 2: Trees

1. [4 points] What are the 5 properties of red-black trees?

2. [8 points] Give on advantage and one disadvantage of each of the four types of trees that we have studied: BST, AVL, Red-black, and Splay. Note that you can not use a single fact two ways. For example, if tree x is faster than tree y, then you can only use that once (i.e., you can say that as an advantage of tree x, but you can not also say that tree y is slower than x as a disadvantage of y).

	Advantage	Disadvantage
BST		
AVL		
Red-black		
Splay		

Page 3: Hashes

3. [8 points] Consider the four hash tables below, each with a different collision resolution strategy. The primary hash function is $h(x) = x \bmod 10$, and the secondary hash function is $h_2(x) = (x \bmod 8) + 1$; note that $h_2(27) = 4$. Insert the following values into *each* of the hash tables: 47, 38, 11, 27.

	Linear probing	Quadriatic probing	Double hashing	Separate chaining
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				

4. [4 points] What are the three necessary properties for a good hash function? Which are absolutely required, and which are necessary for good performance?

Page 4: IBCM

5. [12 points] Given a two-dimensional array of size 10x10 in row-major order somewhere in memory, write IBCM code to compute the instruction to load, into the accumulator, the value at location a[r][c].

   You may assume:

   - the array is zero-indexed, just like C/C++
   - all cells are 1 memory space in size
   - that 'a', 'r', and 'c' are all defined variables
   - there is a multiply function (see below)
   - that you need to store the final instruction at the spot with label 'doit'
   - you may define any other obvious memory values ('ten', for example) that you wish

   To call the multiply function, load the two parameters into memory spots 'p1' and 'p2', and then call 'brl multiply'. Upon return, the result will be stored in memory spot 'ret'.

   Your code should be in opcodes, NOT in hex!

Page 5: x86

6. [3 points] What are the rules for addressing memory in x86? Meaning, how can you combine registers, etc., to specify a memory address?

7. [3 points] When a subroutine is compiled into assembly, why does C++ use a different naming convention than C?

8. [3 points] Why can you not access memory twice in a single instruction? For example, why is `add [var] [eax]` invalid?

9. [3 points] The `push` and `pop` instructions can be implemented using other instructions that we have learned. Give such an alternate implementation, in x86 opcodes, for both of those instructions. In other words, write x86 opcodes that have the same effect as `push` and `pop`. Assume the register being pushed or popped is eax.

Page 6: Miscellaneous

10. [3 points] Why is there no little-theta?

11. [3 points] Give an example how you would use limits to determine the big-Theta (or big-Oh, or big-Omega, etc.) running time of a function. Only one example is needed here.

12. [6 points] Given the following expression in in-fix notation, draw an expression tree, and determine the pre-fix and post-fix versions of this expression. The expression is $(1+7)*(8-(4/2))$, and it evaluates to 48.