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!

Answers for the short-answer questions should not exceed about 20 words; if your answer is too long (say, more than 30 words), you will get a zero for that question!

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.

_____

_____

_____

_____

Three things are certain:
Death, taxes, and lost data.
Guess which has occurred.

(the bubble footer is automatically inserted into this space)

Page 2: Trees

1. [3 points] Produce a list of 7 numbers that, when inserted into an empty binary search tree in the order you list them, produces a perfect binary tree. Recall that a perfect binary tree means that all the leaves have the same depth.

2. [3 points] Show your binary search tree from above after removing the value at the root node.

3. [3 points] Describe three situations where you would *not* want to use a tree data structure.

4. [3 points] In an AVL tree, the left and right sub-trees of a node can vary by at most one. In a red-black tree, by how much can the left and right sub-trees vary by? Briefly explain.

Page 3: Trees and Hashes

5. [3 points] Why should we not use a vector as the secondary data structure when implementing separate chaining for our hash table?

6. [3 points] We can tell when a double rotation is needed by looking at the structure, or knowing that an insert is into the right child of a left child. But how would an AVL insertion algorithm determine this?

7. [6 points] What are the three ways to handle deletes in a hash table? Briefly give an advantage and a disadvantage of each. Note that you can only use a reason once – for example, if x is faster than y, you can't use that as both an advantage of x *and* as a disadvantage of y.

Way to handle delete	**Advantage**	**Disadvantage**

Page 4: Hashes

Questions 8-10 use the hash table to the right, where the primary hash function is $h(k) = k\%10$ (where % is the modulus operator). The values listed there have already been inserted into the table.

0	50
1	
2	32
3	93
4	
5	
6	76
7	
8	
9	19

8. [3 points] If the collision resolution strategy is linear probing, find a value that you could insert that would cause at least *two* unsuccessful probes (the initial hash value and the next attempt at probing). Where does this value eventually end up?

9. [3 points] Now assume that the collision resolution strategy is quadratic probing (and that the value inserted in the linear probing question, above, is not in the table). Find a value that you could insert that would cause at least *two* unsuccessful probes. Where does this value eventually end up?

10. [3 points] Lastly, assume that the collision resolution strategy is double hashing (and that the values inserted above are not in the table). Find a value that you could insert that would cause at least *two* unsuccessful probes. Define the secondary hash function that you use. Where does this value eventually end up?

11. [3 points] What is the load factor of the table after one additional insert (i.e., after one item is inserted into the hash table as seen in the figure above)?

Page 5: IBCM and x86

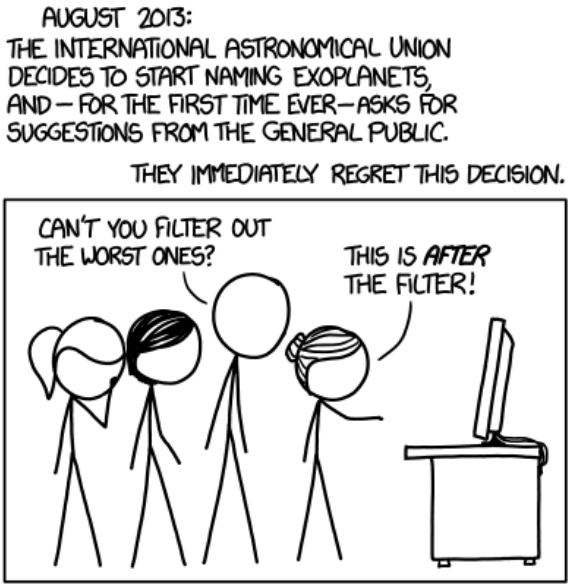
12. [3 points] Using what you know about the calling convention, the stack, and activation records, briefly explain how passing by reference works in x86. How does this allow you to change the value of an actual parameter?

13. [3 points] What are the six things the x86 calling convention places on the stack, in order?

14. [3 points] Briefly explain why parameters are pushed onto the stack in reverse order.

15. [3 points] What change(s) would you have to make to IBCM in order to be able to implement *any* algorithm?

Page 6: Comic!



STAR	PLANET	SUGGESTED NAME
GUESE 667C	b	SPACE PLANET
	c	PILF
	d	A STAR
	e	e'); DROP TABLE PLANETS;--
	f	BLOGOSPHERE
	g	BLOGODROME
	h	EARTH
TAU CETI	b	SID MEIER'S TAU CETI B
	c	GIANT DOG PLANET
	d	TINY DOG PLANET
	e	PHIL PLAINET
	f	UNICODE SNOWMAN
GLIESE 832	b	ASSHOLE JUPITER
GLIESE 581	b	WAIST-DEEP CATS
	c	PLANET #14
	d	BALLDERAAN
	e	ETERNIA PRIME
	f	TAUPE MARS
	g	JELLY-FILLED PLANET
EPSILON ERIDANI	b	SKYDOT
	c	LASER NOISES
GLIESE 176	b	PANDORA
	c	PANTERA
KEPLER-61	b	GOLDENPALACE.COM
UPSILON ANDROMIDAE	c	STAMPY
	d	MOONCHILD
	e	HAM SPHERE
HD 20794	b	COSMIC SANDS
	c	LEGOLAND
	d	PLANET WITH ARMS
HD 85512	b	LAX MORALITY
HD 40307	b	GOOD PLANET
	c	PROBLEMLAND
	d	SLICKLE
	e	SPARE PARTS
	f	NEW JERSEY VI
	g	HOW DO I JOIN THE IAU
GLIESE 163	b	NEIL TYSON'S MUSTACHE
	c	HELP@GMAIL.COM
	d	HAIR-COVERED PLANET
PI MENSAE	b	MOON HOLDER
HD 189733	b	PERMADEATH
KEPLER-22	b	BLUE IVY
KEPLER-3284	b	BLAINSLEY
KEPLER-3255	b	UNICORN THRESHER
KEPLER-2418	b	SPHERICAL DISCWORLD
KEPLER-1686	b	EMERGENCY BACKUP EARTH
KEPLER-3010	b	FEEEOOOOOOOOP
KEPLER-4742	b	LIZ

Figure 1: "Exoplanet Names" (http://xkcd.com/1253/)