

CS 2150 Exam 2, spring 2015

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.

*A crash reduces
Your expensive computer
To a simple stone.*

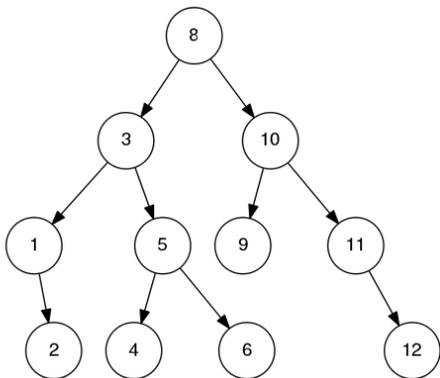
(the bubble footer is automatically inserted into this space)

Page 3: Trees

5. [3 points] Why would we never implement a separate chaining bucketing using red-black trees? Also, why not with vectors?

6. [3 points] We say that the three primary operations on a splay tree have $\Theta(\log n)$ *amortized* running time. What does the *amortized* part of that mean?

7. [3 points] Insert 7 into the AVL tree shown below. Show the resulting tree.



8. [3 points] When would you want to use an AVL tree over a red-black tree?

Page 4: Hashes

9. [3 points] Give two reasons why a hash table size must always be prime.
10. [3 points] How can you reverse an MD5 hash?
11. [6 points] Give one advantage and one disadvantage of each of the collision resolution strategies that we have studied in this course. Note that you can't use the same reason twice! So if A is faster than B , you can't *also* say that B is slower than A .

	Advantage	Disadvantage
Separate chaining		
Linear probing		
Quadratic probing		
Double hashing		

Page 5: x86

12. [12 points] The x86 code, below, is supposed to implement the same functionality as the provided C++ function. Fill in the missing x86 instructions. Assume `product()` is the function that you implemented in pre-lab 8.

```
int f1 (int x, int y, int z){
    int w = x+y;
    while (z<6) {
        z++;
        x+=2;
    }
    return w*(x+z)-y;
}
```

The C++ code above is translated into the following assembly. In order for this to be complete, you have to fill in the blank lines:

<pre>f1: push ebp mov ebp, esp sub esp, 4 push ebx push esi push edi _____ mov ebx, [ebp+12] mov edx, [ebp+16] _____ add esi, ebx add esi, ecx mov [ebp-4], ebx mov ebx, 2 mov edi, 6 L1: cmp edx, edi jge L2 inc edx add ecx, ebx</pre>	<pre>_____ L2: mov eax, ecx add eax, edx _____ push esi call product add esp, 8 mov ecx, [ebp-4] _____ pop _____ pop _____ pop _____ add esp, 4 pop ebp ret</pre>
--	---

