CS 2150 Final Exam

Name

You MUST write your e-mail ID on **EACH** 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 – sorry to have to be strict on this!

There are 8 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 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.

Page 2: C++ and Linked Lists

1. [12 points] For this page, you will examine the implementation of a Singly Linked-List in C++ (shown below). The code presented has exactly 4 syntax errors (i.e., issues that prevent compilation) and 4 logic errors (i.e., the linked-list will not operate as expected). For each line that contains one of these 8 errors, fix it by either crossing it out and replacing with a working line of code OR draw an arrow to move that line of code elsewhere. You can fix this code by correcting exactly 8 different lines (one per error). No fix will require extra lines of code nor will one line contain both a syntax and logic error.

```cpp
LinkedList :: LinkedList (){
  //dummy head node
  head = new ListNode ();

  head.next = NULL;

  size = 0;

}

LinkedList ::~ LinkedList (){

  makeEmpty ();

  delete head;

}

void LinkedList :: insertAtHead (int x){

  ListNode nn = new ListNode ();

  nn->value = value;

  head->next = nn;

  nn->next = head->next;

  size ++;

}
```

```cpp
//remove all except dummy head
void LinkedList :: makeEmpty (){

  while (size >= 0)

    removeHeadNode ();

}

bool LinkedList :: find (int x){

  ListNode* cur = head;

  while (cur != NULL) {

    if (cur->value == x) return true;

    cur->next = cur;

  }
  return false;

}

void LinkedList :: removeAtHead (){

  toDel = head->next;

  head->next = toDel->next;

  delete toDel;

  size --;

}
```

Page 3: Runtimes

2. [12 points] For each data structure in the table below, state the *Big-Theta* worst-case runtime for each operation unless otherwise stated. For some structures, you will need to replace insert, delete, or find with the correct intuitive corresponding operation for that structure (e.g., push() on a stack instead of insert()). We've clarified a few of the less obvious examples of these in the assumptions below, which contains other assumptions you should make for your analyses as well.

Structure	Assumptions	find()	insert()	delete()
Vector	Use amortized time; insert/delete at end of array			
Linked List	insert/delete at tail			
Stack	Vector Based; Use amortized time; find() is top()			
Queue	Array Based; Use amortized time; find() is look at next item to dequeue			
Bin. Search Tree	Worst case			
AVL Tree	Worst case			
Red-Black Tree	Worst Case			
Splay Tree	Use amortized time			
Hash Table	Linear probing; do not consider array resize; **worst-case**			
Hash Table	Linear probing; do not consider array resize; **best-case**			
Priority Queue	Uses a sorted vector; do not consider array resize			
Priority Queue	Binary Heap; use expected time; find() is findMin()			

Page 4: Numbers

3. [4 points] Convert the following numbers into or out of *8-bit twos-complement binary* in order to complete the chart.

Twos-Complement Binary	Twos-Complement Hexadecimal	Decimal
10011011	0x9B	
11111010		
		-128
		89

4. [4 points] This question is about converting into IEEE Floating Point notation. Suppose we have a float notation similar to IEEE 754 notation, except there is 1 sign bit, 3 exponent bits, and 4 mantissa bits (*Note that the exponent offset is* $110_2 - 011_2 = 011_2 = 3_{10}$). Convert to and from this format to complete the chart below.

	Floating Point Binary	Decimal
Number 1:	11010101	
Number 2:		$0.375\ (\frac{3}{8})$

5. [4 points] What is the exact smallest, finite, positive, non-zero number this format can represent? Recall that exponent values of zero are special values and cannot be used here.

Page 5: x86

6. [8 points] For each x86 register below, list its usage in the *Calling Convention*. Your options include return value, parameter n (give specific value for n), caller saved, callee saved, or list a special usage for that register (you should ONLY write one of those options).

Register	Usage	Register	Usage
RAX		R8	
RBX		R9	
RCX		R10	
RDX		R11	
RSI		R12	
RDI		R13	
RSP		R14	
RBP		R15	

7. [4 points] Look at the C++ code below and translate the code into x86 by filling in the missing lines of code.

```
//x:pointer to array; n:array length; c:current position for sum (starts at 0)
int recursiveSumArray(int* x, int n, int c){
    if(c == n) return 0;
    return x[c] + recursiveSumArray(x, n, c+1);
}
```

```
recursiveSumArray:                              mov r10, rdi



    _____              _____

    je base                             add rax, r10
                                        jmp done
                                    base:


    _____
                                    _____
    inc rdx
    call recursiveSumArray
    pop rdx                             done:
                                            ret
```

Page 6: Heaps and Huffman

8. [6 points] Draw the min-heap (in tree form) that results from inserting the following integers in the following order: 5, 12, 10, 1, 3, 2

9. [3 points] Draw the *Huffman Tree* that is produced by the following uncompressed file (ignore the spaces, they were added to make the string easier to read): a b b c c c d d d d e e e e e e

10. [3 points] List three advantages of storing a min-heap as an *array*, instead of using nodes and pointers.

Page 7: Graphs

11. [3 points] Consider the graph below, which has two unique *Minimum-Spanning Trees*. Draw out each of the two MSTs for this graph.
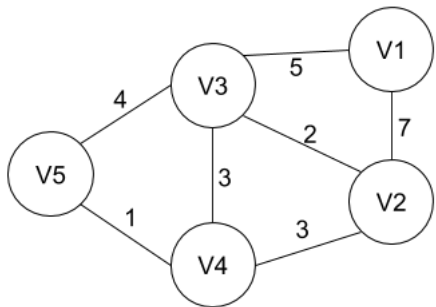
12. [3 points] TRUE or FALSE: If a graph has all unique edge weights, then the Minimum-Spanning Tree must be unique. If true, provide your high level reasoning. If false, draw a small graph that serves as a counter-example.

13. [6 points] Perform Djikstra's Algorithm on the graph at the top of this page. Use V_1 as the start node. Fill in the table below. When updating a value in a cell, place a single line through the old value and show the new value adjacent to the old one.

Node	Known	Dist	Path
V_1		0	
V_2			
V_3			
V_4			
V_5			

Page 8: Demographics　　　　　**Name & userid:** _____

We meant to ask these in an end-of-the-semester survey, but we did not get to it in time. So we'll put it here for some extra points on the exam! Sorry if this page is a bit crowded…

14. [0 points] Did you put your name and userid at the top of this page? You need to do so in order to get the points on this page!

15. [2 points] What is your major or minor? If you have not declared, then answer with your intended major or minor. Please circle one.

    - BS CS
    - BA CS
    - BS CpE
    - CS minor
    - Other (please explain): _____
    - Neither majoring nor minoring in computing

16. [1 points] Have you already declared the major/minor mentioned above? Circle: Yes or No

17. [2 points] What CS 1 class did you take? Please circle one.

    - CS 1110
    - CS 1111
    - CS 1112
    - CS 1120
    - AP credit
    - Transfer credit
    - Other (please explain): _____
    - Placed out of it via the CS 1110 placement exam

18. [1 points] If you took your CS 1 class in college (i.e. CS 1110, CS 1111, CS 1112, CS 1120, or a transfer class), in what semester did you take it? Please specify a semester by season and calendar year (i.e., "fall 2014" and not "my second year").

19. [2 points] What CS 2 class did you take? Please circle one.

    - CS 2110
    - CS 2220
    - AP credit
    - Other (please explain): _____
    - Transfer credit
    - Placement exam

20. [1 points] If you took your CS 2 class in college (i.e. CS 2110, CS 2220, or a transfer class), in what semester did you take it? Please specify a semester by season and calendar year (i.e., "fall 2014" and not "my second year").

21. [1 points] Did you attend the final exam review session? You'll get full credit for this question, as long as you answer it honestly (we know most that were there, but not all).

22. [2 points] For the 3-credit courses for next semester (not summer or J-term):

    - How many CS courses are you enrolled in (not wait-listed)?

    - How many CS courses are you wait-listed for?

    - How many CS courses would you *like* to be enrolled in?