CS 2150 Exam 2

You MUST write your name and e-mail ID on EACH page and bubble in your userid at the bottom of EACH page – including this page.

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 forms. So please do that first. Sorry to have to be strict on this.

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

There are 10 pages to this exam – once the exam starts, please make sure you have all 10 pages.

The first and last pages are worth 2 points each (you get that if you bubble in your userid). The other 8 pages are worth 12 points each. Questions are worth different amounts, depending on the question length. The three and four point questions on this exam should not take more than a line or two to answer – **your answer should not exceed about 20 words.** There are a total of 100 points of questions on this exam, and 1 hour 45 minutes (105 minutes) to take the exam, which means you should spend about one minute per question point – the remaining 5 minutes are to fill out the bubble footers.

If you do not bubble in a page, you will not receive credit for that page!

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.

____

____

____

____

Out of memory.
We wish to hold the whole sky,
But we never will.

Exam 1 Material

1. [4 points] Briefly give a convincing example of when we would want to allow implicit constructor initialization in a C++ class.

2. [4 points] Formally prove that $3n$ is $O(n^2)$

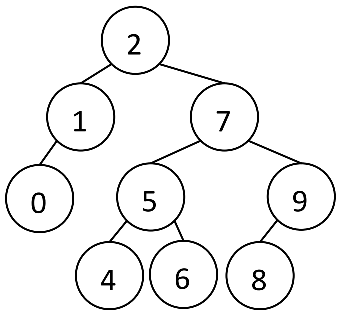
3. [4 points] Consider the declaration `int x[4][2]`. Diagram how this is represented in memory. In particular, it must be clear (via your diagram or an English explanation) where the 2-dimensional cells map to your memory diagram.

Trees

4. [3 points] Write the pseudo-code (or an English explanation) for the binary search tree remove. We're looking for a high-level overview here!

5. [3 points] Write pseudo-code (or an English explanation) for the splay operation. We're looking for a high-level overview here!

6. [3 points] Insert 3 into the AVL tree below. Show the resulting data structure.



7. [3 points] Briefly, what are the 5 properties of red-black trees?

Hashes

8. [6 points] Consider the four types of collision resolution that we have seen: separate chaining, linear probing, quadratic probing, and double hashing. BRIEFLY give one advantage and one disadvantage of each of these types of collision resolution.

Collision resolution method	**Advantage**	**Disadvantage**
Separate chaining		
Linear probing		
Quadratic probing		
Double hashing		

9. [3 points] What are the desired properties of a good hash function?

10. [3 points] Give two reasons why a hash table should be a prime number in size, and BRIEFLY explain each reason.

IBCM

11. [12 points] Write a complete IBCM program that reads in three values from the keyboard: *a*, *i*, and *x*. *a* is an array base address, and *i* is an index into that array. The program needs to store *x* at *a*[*i*]. You can leave your answer in IBCM opcodes (i.e. 'add one', where 'one' is a label). However, all variables MUST have their initial values clearly specified. The IBCM commands (and their hex values) are listed to the right.

IBCM

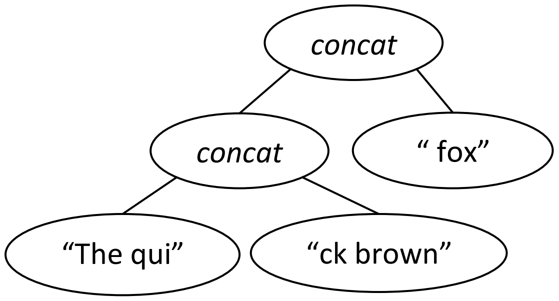
0	halt
1	I/O
2	shifts
3	load
4	store
5	add
6	sub
7	and
8	or
9	xor
A	not
B	nop
C	jmp
D	jmpe
E	jmpl
F	brl

(the bubble footer is automatically inserted in this space)

Ropes

Consider the data structure called a *rope*. This data structure is used to contain a sequence of characters, similar in function to a string – hence its name. It has the same basic operations as a string, four of which we care about for this exam: concatenation (putting two strings/ropes together), substring (finding a substring/subrope within the overall string/rope), indexing (finding a character based on its index in the string/rope), and iteration (going through each character in the string/rope). Indeed, ropes are a common data structure, and are available in the C++ STL.

A rope keeps its data stored using a binary tree. Every leaf node is a sequence of characters, possibly as few as just one character. Every internal node is the concatenation operation. The rope shown to the right is equivalent to the string "The quick brown fox".



12. [4 points] Consider each of the four operations described above (concatenation, substring, indexing, and iteration). For a *standard string kept in an array*, what is the running time for each?

Operation	**Running time**
Concatenation	
Substring	
Indexing	
Iteration	

13. [8 points] BRIEFLY describe how you would perform each of the four operations on a rope. If you answers are not brief, we will take points off!

(the bubble footer is automatically inserted in this space)

Ropes

14. [3 points] For the rope as presented on the previous slide (using a binary tree), what is the running time for each of the same operations?

Operation	**Running time**
Concatenation	
Substring	
Indexing	
Iteration	

15. [3 points] Using the self-balancing trees that we studied, how could we improve the performance of a rope?

16. [3 points] With the improvement you described in the previous question, what would the running time of the four operations be?

Operation	**Running time**
Concatenation	
Substring	
Indexing	
Iteration	

17. [3 points] Some of the advantages and disadvantages of ropes are their running times, which you answered in the previous question. However, there is at least one advantage and one disadvantage concerning memory for a rope. What is the advantage and what is the disadvantage?

---

x86

18. [9 points] Draw a complete x86 activation record for a subroutine that takes in two parameters. We don't care if you have your registers mixed for this, since it's a fairly new topic. For each 'optional' step (backing up a register, etc.), you can assume that at least one 'thing' is done (i.e. at least one register is backed up in that step).

19. [3 points] It's 2 a.m., and I'm tired and don't feel like trying to think up of another exam question. So write down 3 x86 assembly opcodes and what they do, and you'll get credit for this question.

Miscellaneous

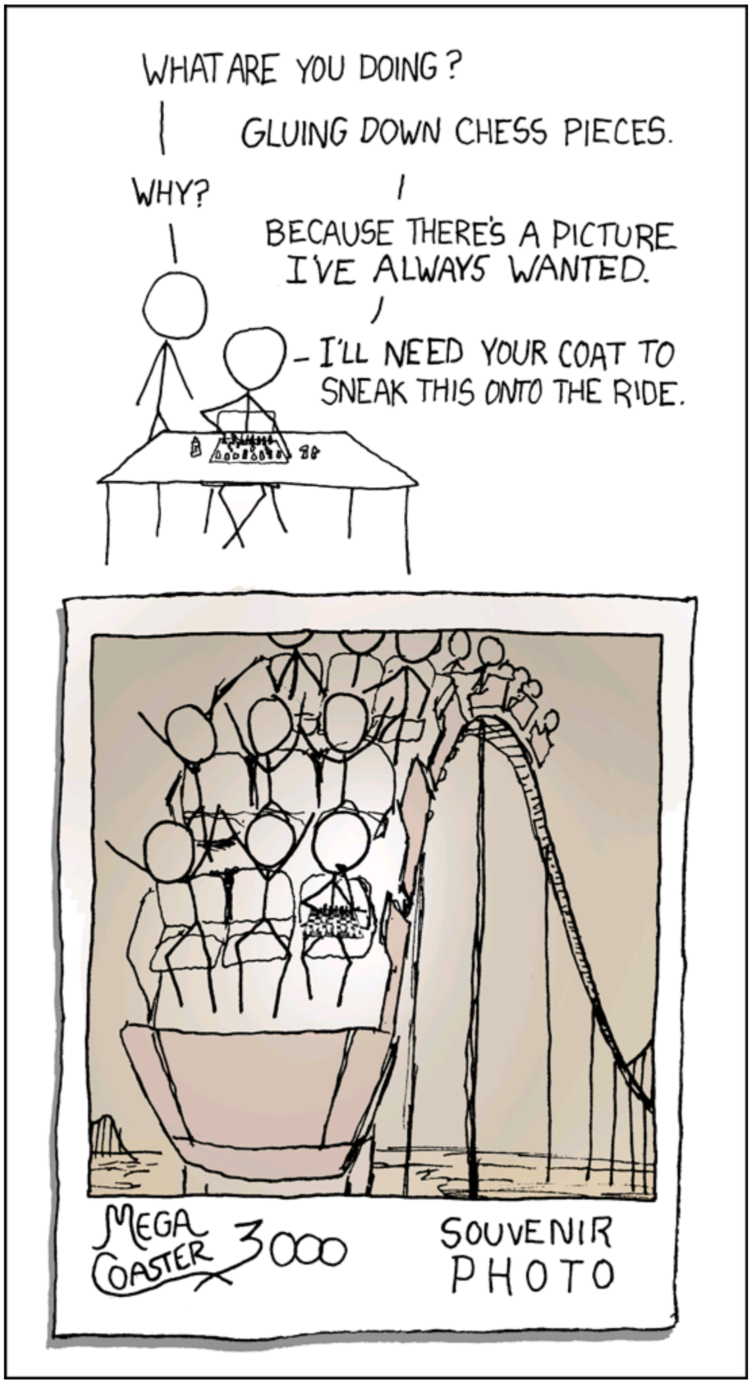
20. [3 points] What audio recording did Professor Bloomfield play in class yesterday (Mon, March 29th)?

21. [3 points] List all the Emacs keyboard commands that you know, and give a BRIEF description of what each one does (a word or two is fine here). At this point in the semester, you should know at least 8.

22. [6 points] Write a bash shell script that will read in two variables, D and G (you can call them anything). It will then run a given program (we'll say a.out) three times, passing those inputted values as command-line parameters. The last line of each of those program executions is a number – those 5 numbers (the last line from each of the 5 executions) will be averaged, and that average printed out. You do not need loops or if-else statements for this.

This page unintentionally left blank.

But you get two points for bubbling in the form at the bottom of this page. Woo-hoo!


WHAT ARE YOU DOING?
GLUING DOWN CHESS PIECES.
WHY?
BECAUSE THERE'S A PICTURE I'VE ALWAYS WANTED.
I'LL NEED YOUR COAT TO SNEAK THIS ONTO THE RIDE.
MEGA COASTER 3000
SOUVENIR PHOTO

(the bubble footer is automatically inserted in this space)