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 10 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.

You answers should be brief – if they are too long (longer than, say, 20-30 words), you will receive zero credit for that question! This is not an essay exam. We don't want you counting words in your answers, but we want you to keep the length reasonable.

_____

_____

_____

_____

A crash reduces
Your expensive computer
To a simple stone.

(the bubble footer is automatically inserted into this space)

Page 2: Lists

1. [4 points] List two benefits and two drawbacks for implementing a queue data type via arrays.

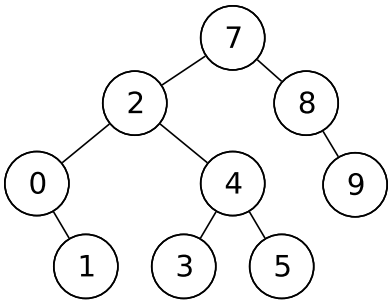
2. [4 points] What is an abstract data type (ADT)? Why would we use them?

3. [4 points] In the `vector` class, if the vector's internal array (that holds the values) fills up, then the vector has to double the size of that array. How does this occur in C++, when array base names are constants and array lengths are fixed?

Page 3: Trees

4. [4 points] Splay trees have a $\Theta(\log n)$ amortized running time for the primary operations. What does the *amortized* mean? Give an example of another data structure with an amortized running time.

5. [4 points] Insert 6 into the AVL tree below. Show the resulting data structure.



6. [4 points] Explain the structure of a red-black tree. We aren't asking for a recitation of the red-black tree rules, but instead a description of what structure those rules enforce on red-black trees.

Page 4: Hashes

7. [3 points] What is the running time of an insert into a hash table, assuming a probing strategy for collision resolution?

8. [3 points] Of the three properties for a good hash function, which are required for a correct hash table implementation, which are only desirable (i.e., not strictly required)? Give a brief explanation for each.

9. [3 points] Give a benefit and a drawback of a small load factor (such as 0.25); also give a benefit and a drawback of a high load factor (such as 1.0).

10. [3 points] How can we have good hash table performance when we have a "slow" data structure (such as a linked list) as the separate chaining bucket, instead of a "fast" data structure (such as an AVL tree)?

Page 5: IBCM

11. [4 points] What is the purpose of cache in the memory hierarchy?

12. [4 points] Write an IBCM code snippet (i.e., just the relevant code, not an entire program) that will perform a `while` loop while variable `B` is greater than or equal to 5. Don't worry about the body of the `while` loop – we are interested in the loop control part, not the body. Your answer should be in IBCM opcodes; we don't want to be looking at hexadecimal notation. For any variable locations, you can ignore exactly where they are in memory, and just use the variable name.

13. [4 points] Write an IBCM code snippet that will perform one thing if a variable B greater than 7, and something else if it is not 7. Again, we just want the relevant code that controls the `if` statement. And it should be in IBCM opcode format, not hexadecimal.

Page 6: x86

14. [3 points] Why does C++ use a different naming convention when assigning a name in assembly to a subroutine call in C++? In other words, if you name a subroutine `foo()` in the C++ code, it names it something quite different in the assembly – why is this the case?

15. [3 points] In the callee's prologue, there is a `mov ebp, esp` line, and a corresponding `mov esp, ebp` line in the prologue. What do these lines do? We want an explanation of what happens, not a literal translation of the opcodes.

16. [3 points] What are all the parts of an activation record? Meaning, what is put onto the stack? List them in order that they are pushed onto the stack!

17. [3 points] Explain exactly why the parameters must be pushed onto the stack in *reverse* order.

Page 7: UNIX

18. [3 points] What does the `chmod` command do? In your answer, it must be clear *why* we would need to do whatever the `chmod` command does.

19. [3 points] Briefly explain what the UNIX pipes do. There are four of them: '<', '>', '>>', and '|'.

20. [3 points] Briefly describe the following terms for Makefiles, and what they do: suffix rules, dependencies, and targets.

21. [3 points] Give two examples of what shell scripts are good for, and two examples of what they are *not* good for.

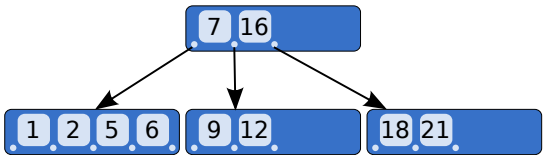
Page 8: x-ary trees

While this is a lot to read, you get a free 8 points for reading it.

An x-ary tree is somewhat similar to a binary tree, but it can hold up to x values per node, and up to $x + 1$ children per node; thus, a binary tree would be a 1-ary tree. In x-ary trees, x can be in the hundreds or thousands.

All values in a node are kept in sorted order; thus, a 4-ary tree would have values v_1, v_2, v_3, v_4 in each node, where $v_1 < v_2 < v_3 < v_4$ (like a binary tree, we'll assume that you can not have duplicate values). Such a node will have 5 children, one between each node. The child between v_1 and v_2 has values that are strictly between v_1 and v_2. And the child to the left of v_1 has values that are all less than v_1.

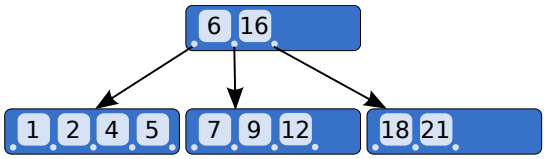
Note that in a x-ary tree, each node can have *up to* x values, but it can have fewer. However, a single node cannot have less than $\lceil x/2 \rceil$ values (the only exception is the root node if the total number of values in the entire tree is less than $\lceil x/2 \rceil$).



The trees shown here are examples of 4-ary trees. In the first tree, note that the child between values 7 and 16 in the root node has values that are all greater than 7 and less than 16.

Like AVL or red-black trees, x-ary trees are balanced. In fact, the balancing ensures that *all* leaves are of the same depth.

Inserts only happen in the leaves. If a leaf is full, then a value gets "pushed" up, and then another value gets "pushed" down into the next leaf over. If we were to insert 4 into the tree above, it would "push" 6 (the highest value in that node) up to the root node, which would cause 7 to be "pushed" down to the center child node. The resulting tree would be what is shown to the right.



Some details of x-ary trees are not described here, as they are beyond the scope of this exam (in particular, details about exactly how the "pushing" works, the balancing of all leaves at the same level, etc.). But rest assured, the details do describe a consistent data structure.

22. [4 points] Compare a 2000-ary tree and a balanced binary tree. Assume, for this question, that moving up or down a level in the tree takes 1 second, but accessing the values in a node is negligible. How long (in seconds!), in the worst case, will it take to find a value in a tree of 10^9 elements? Answer for both types of trees. Note that $2^{10} \approx 10^3$.

Page 9: x-ary trees, page 2

23. [4 points] When would you want to use a x-ary tree instead of a balanced binary search tree? When would you want to use a balanced binary search tree instead of an x-ary tree?

24. [4 points] Compare the running time of balanced binary search trees to x-ary trees. If they are in the same big-Theta running class, you should still explain a differentiation between them.

25. [4 points] Given the provided description of x-ary trees, and in particular the "pushing" of values from one leaf to another, what is the big-Theta estimate of the worst-case running time of an insert? Why?

Page 10: Comics!

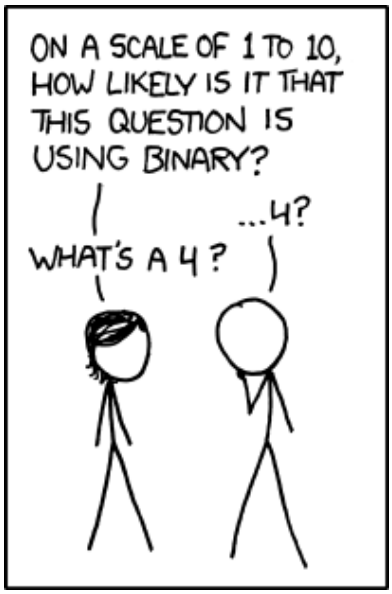


Figure 1: http://xkcd.com/953/

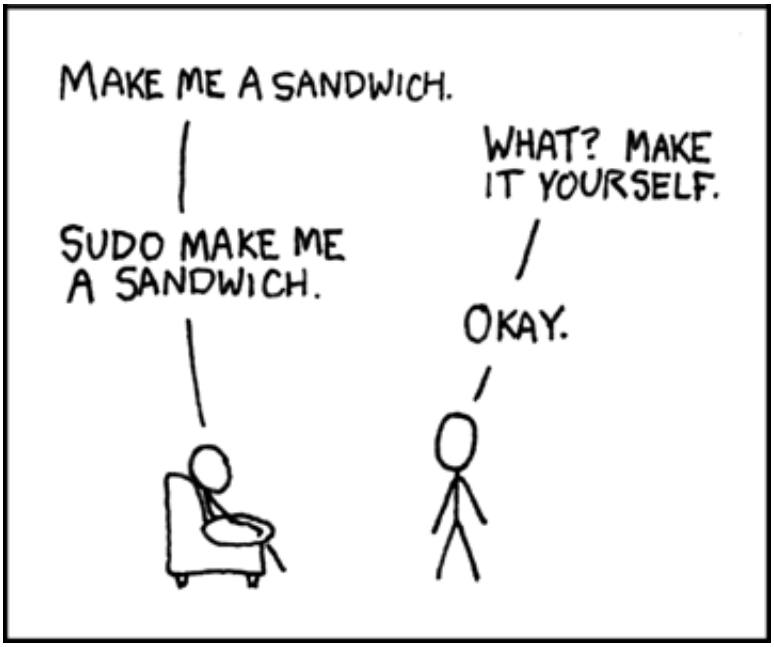


Figure 2: http://xkcd.com/149/ ('sudo' runs a command as the super-user on a UNIX system)