 Name: ____________________________ Email ID: __________

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 do not do this, you will receive a zero for that 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.

Your answers must be completely on the page in which the question was asked (and not in the footer). We can provide you with scrap paper and/or additional test pages, if necessary.

There are 10 pages to this exam – once the exam starts, please make sure you have all 10 pages. Pages 2-10 of this exam contain a combination of short-answer, medium-answer, and long-answer questions. Each page is worth 20 points, for a total of 180 points. Conveniently, there are 180 minutes (3 hours) to this exam – so you should not be spending, on average, more than 1 minute per point of a problem.

Short answer questions (those worth 5 points) should not take more than a sentence or two to answer.

This exam is CLOSED text book, closed-notes, closed-calculator, 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 here:

Name: ____________________________ Email ID: __________

First exam stuff

1. [10 points] Convert -12.0 to a IEEE 754 double-precision floating point number. If you don't remember the mantissa and exponent size, take your best guess. And remember that the exponent offset is always $2^{n-1}-1$, where n is the number of bits in the exponent. Your answer should be in big-endian hexadecimal format. For simplicity, we'll assume that everything after the 5^{th} bit in the mantissa is a 0.

2. [5 points] Why doesn't anybody use 1's complement integers these days?

3. [5 points] Give a big-Oh estimate for the *recursive* running time of fibonacci(n) (here n means the n^{th} term of the sequence)

Name: ____________________________ Email ID: __________

Second exam stuff

4. [5 points] How does double hashing resolve a collision?

5. [5 points] AVL trees have very good performance. Yet they are almost never used – for example, in the Java SDK, the TreeMap interface uses red-black trees instead. Why are AVL trees almost never used?

6. [5 points] What is the fetch-execute cycle?

7. [5 points] What is the big-Oh running time of a hash table's insert method?

a b c d e f g h i j k l m n o p q r s t u v w x y z
a b c d e f g h i j k l m n o p q r s t u v w x y z
a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
a b c d e f g h i j k l m n o p q r s t u v w x y z
a b c d e f g h i j k l m n o p q r s t u v w x y z

Do not write in this area
Department Course Exam Page Version

 Name: ____________________________ Email ID: __________

Complexity

8. [5 points] Give an example of when big-Oh notation breaks down. Meaning, when is the big-Oh analysis not helpful when comparing algorithms?

9. [5 points] Why is there no little-theta?

10. [5 points] We have studied a number of running times that can be considered "polynomial" – meaning anything that runs in O(n^c) time, where c is a constant. What are the running times that we have studied that are "bigger" than polynomial – i.e. exponential? We are looking for the big-Oh times, not the algorithms that are exponential.

11. [5 points] What does it mean when we say that an algorithm is NP-complete?

 Name: ____________________________ Email ID: __________

x86 assembly

12. [5 points] What are the prologue steps of the caller's calling convention?

13. [5 points] What are the prologue steps of the callee's calling convention?

14. [5 points] What are the epilogue steps of the callee's calling convention?

15. [5 points] What are the epilogue steps of the caller's calling convention?

a b c d e f g h i j k l m n o p q r s t u v w x y z
a b c d e f g h i j k l m n o p q r s t u v w x y z
a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
a b c d e f g h i j k l m n o p q r s t u v w x y z
a b c d e f g h i j k l m n o p q r s t u v w x y z

Do not write in this area
Department Course Exam Page Version

 Name: ___________________________ Email ID: __________

Memory

16. [5 points] Why are caches important?

17. [5 points] Write a snippet of C/C++ code that declares an array of *n* ints using malloc(). Note that you do not necessarily know the size of ints on the system you are going to compile this code on.

18. [5 points] What is cache temporal locality?

19. [5 points] What is cache spatial locality? What data structure has the best spatial locality?

a b c d e f g h i j k l m n o p q r s t u v w x y z
a b c d e f g h i j k l m n o p q r s t u v w x y z
a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
a b c d e f g h i j k l m n o p q r s t u v w x y z
a b c d e f g h i j k l m n o p q r s t u v w x y z

Do not write in this area

Department Course Exam Page Version

 Name: ____________________________ Email ID: __________

Heaps

Note: binary heap, heap, and priority queue are all synonymous. And assume we are talking about min-heaps unless specified otherwise.

20. [5 points] What is the structure property for a heap?

21. [5 points] What is the ordering property for a heap?

22. [5 points] What is the most efficient way to store a heap in memory? Why?

23. [5 points] Give two real-world applications of heaps other than Huffman coding. "Real-world" application means that just saying "finding the minimum element" doesn't count here.

 Name: ____________________________ Email ID: __________

Huffman Coding

24. [10 points] Encode the following message using Huffman coding: “to be or not to be”. You will need to encode the spaces as well as the letters. What is the Huffman tree cost and the compression ratio? You can leave these as formulas. Show your Huffman coding tree!

25. [10 points] Decoding the following Huffman encoded message. There are 12 character encodings, given in on the left. Note that the first character is a colon (‘:’), and the second character is a space (‘ ‘). The encoded message is on the right. What is the Huffman tree cost and the compression ratio? You can leave these as formulas. Show your Huffman coding tree!

: 1000	e 1011	s 010	11010010101010101101111110
111	n 1001	t 0000	00000101110001000110001111
2 0001	o 0010	w 1010	1100110111100011010
a 110	r 011	y 0011	

 Name: ____________________________ Email ID: __________

Graphs

26. [5 points] What is a topological sort? When would we use it?

27. [5 points] Describe (concisely, but accurately) how Dijkstra's single source shortest path algorithm works.

28. [5 points] Describe (concisely, but accurately) one of the two algorithms for finding the minimal spanning tree in a connected graph (Prim's or Kruskal's). For full credit, specify which one you are describing.

29. [5 points] The current record for finding a (unique) solution to the travelling salesman problem is a simulation in April 2006 that found the shortest cycle among 85,900 vertices. Why is solving such a inefficient problem (the brute force algorithm is O(n!)) in a (relatively) short amount of time so important?

 Name: ____________________________ Email ID: __________

Data structures

30. [20 points] Enter the big-Oh running times for each of the data structures for each operation in the table below. See the notes below the table.

Data structure	Insert	Remove	Remove min	Find	Print all
Unsorted list (array)					
Unsorted list (linked)					
Sorted list (array)					
Sorted list (linked)					
Stack					
Queue (normal)					
BST					
AVL tree					
Splay tree					
Red-black tree					
Hash table					
Heap / priority queue					

The “remove” operation should be interpreted as appropriate for the data structure. For example, removing from a list means removing a specific element; removing from a stack means removing the last pushed element (i.e. pop()). Likewise with the other operations (insert is push() for a stack, etc.). If an operation is not possible or not intended with the given data structure, please enter “N/A” – for example, we can’t really remove the minimum element from a stack, but we can from a list. In other words, interpret each operation as appropriate for the given data structure.