CS161 - Final Exam
Computer Science Department, Stanford University
August 16, 2008

Name: ______________________________

Honor Code

1. The Honor Code is an undertaking of the students, individually and collectively: a) that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading and b) that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.
2. The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.
3. While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

Signature: ______________________________

1	2	3	4	5	6	7	8	Total:
20	40	15	15	15	20	20	12	157

1. True/False (20 points) - State whether the statements below are true or false and give a brief justification for your answer. 1 point for T/F and 1 point for justification.

   ______ The recurrence $T(n) = T(n-1) + n \log n$ is $O(n^2 \log n)$

   ______ Running BFS and DFS on a tree of depth 1 produces the same traversal order

   ______ For any positive weighted graph, the Bellman-Ford algorithm gives the shortest path from a vertex v to all other vertices

   ______ $f(n) = O(g(n))$ implies $g(n) = O(f(n))$

   ______ For building an FSA string matcher, the key idea is to find the largest prefix of the state plus transition character (e.g qa) that is also a suffix of the pattern.

   ______ The reason B-Trees have a large branching factor is because they are designed for use on disk

   ______ $f(n) = O(g(n))$ implies $2^{f(n)} = O(2^{g(n)})$

   ______ If a polynomial time algorithm exists to detect a HAMILTONIAN-CYCLE, then a polynomial time algorithm exists to solve the SUBSET-SUM problem

   ______ The main way to identify a greedy algorithm is to look for overlapping optimal subproblems

   ______ Amortized analysis is a tool for evaluating the average running time over a set of operations, rather than the cost for an individual operation

2. (40 points - 8 each) Short answer

   (a) Under a huffman encoding of n symbols with frequency $f_1, f_2, ..., f_n$, what is the longest a codeword could be? Give an example of a set of frequencies that would produce this case.

   (b) Given a graph $G = (V, E)$, you are asked to design the best possible algorithm for the problems below. Give the worst-case complexity for these algorithms for a graph represented as both an adjacency matrix and adjacency list in terms of V and E. (You do not need to specify the algorithm)

   - How many vertices in an undirected graph have exactly two edges?

     matrix: ______________ list: ______________

   - Are there no edges at an in an undirected graph?

     matrix: ______________ list: ______________

   - Is there a path from vertex 1 to vertex $|V|$ in a directed graph?

     matrix: ______________ list: ______________

(c) Suppose that you need to insert 100,000 keys into a hashtable with 200,000 buckets. You are very unlucky and all of the keys hash to the same bucket. Would you be better off using resolution by chaining, linear probing or double hashing? Explain your answer.

(d) Prove that it is not possible to implement a priority queue so that an n-element priority queue can be created in $O(n)$ time and the largest element can be extracted (and deleted) in $O(1)$ time.

(e) Is it possible to design an $O(n \log n)$ sorting algorithm whose only means of rearranging elements is by exchanging adjacent elements? Explain your answer.

3. (15 points) Shortest paths

   Given a directed graph $G = (V, E)$ with positive edge weights and a particular node $v_i \in V$, give an efficient algorithm for finding the shortest paths between **all pairs of nodes**, with the one restriction that these paths must all pass through v_i. Give the runtime of your algorithm. Points will be deducted for an inefficient algorithm.

4. (15 points) 2-Sum

   Given an input array A of n distinct positive integers and a positive integer x, describe an algorithm to determine whether or not there are two integers in A that sum to x. Your algorithm should run in time $O(n \log n)$.

5. (15 points) Vertex cover

   A *vertex cover* of a graph $G = (u, v)$, is a subset $V' \subseteq V$ such that if $(u, v) \in E$ then $u \in V'$ or $v \in V'$ (or both). Given an undirected tree $T = (V, E)$, you're asked to give an algorithm that outputs the *smallest* vector cover of T.

   Is this problem NP-Complete? If so, provide a proof. If not, provide a polynomial time solution. Points will be deducted for an inefficient algorithm.

6. (20 points) DOUBLE-SAT

   DOUBLE-SAT is the following problem: Given a boolean formula of n boolean variables $x_1, x_2, ..., x_n$ joined by m boolean connectives (one of: $\wedge$ (AND), $\vee$ (OR) and $\neg$ (NOT)), are there **two** different assignments of the variables such that the boolean formula evaluates to 1? For example, $(x_1 \vee \neg x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_3)$ has two valid assignments, $x_1 = 1, x_2 = 1, x_3 = 0$ and $x_1 = 0, x_2 = 1, x_3 = 0$

   Prove that DOUBLE-SAT $\in$ NP-Complete

7. (20 points) Scheduling

The manager of the student union comes to you with the following problem: she's in charge of a group of n students, each of whom is scheduled to work one shift during the week, where a shift is defined by a start time s and an end time f. There may be multiple shifts going on at once.

She's trying to decide on a subset of these n students to compose a supervising committee. She considers such a committee to be complete if, for every student not on the committee, that student's shift overlaps (at least partially) with the shift of some student who is on the committee. Thus, each student's performance can be observed by at least one person who serves on the committee.

For example, suppose $n = 3$ and the shifts are
monday 4pm-8pm ($s_1 = 16$, $f_1 = 20$)
monday 6pm-10pm ($s_2 = 18$, $f_2 = 22$)
monday 9pm-11pm ($s_3 = 21$, $f_3 = 23$)

The smallest supervising committee consists of just the 2nd student.

Give an efficient algorithm that takes the schedule of n shifts and produces a complete supervising committee containing as few students as possible and prove that this algorithm gives the minimum number of supervisors. Points will be deducted for an inefficient algorithm.