CS180 Final Exam
Winter 2011

Name	
ID	

DO NOT OPEN EXAM UNTIL INSTRUCTED TO DO SO

This final exam is closed book and closed notes. It is expected to be an individual effort; you may not communicate with anyone other than the designated proctors *for any reason*, nor may you use any communication devices during the exam. You may not have any materials available to you other than those used for writing (pencil, eraser, etc) and your photo ID.

You will have three hours, from 11:30 AM until 2:30 PM, to work on this exam. If you are still working at 2:20 PM, you will not be permitted to turn in your exam until the end, out of consideration for those who may still be working. If you have any questions during the exam, please raise your hand to get the attention of the proctor.

When grading this exam, the only portions of your exam that will be graded for credit on question i are on the front and back of the page that contains question i. However, we will keep this packet intact; you need only include your name on the front of the packet. There are extra pages at the back of this packet that may be used for scratch paper. Nothing on those pages will be graded. We suggest that you determine what your answer will be using scratch pages before writing your answer on the to-be-graded pages. If you need additional scratch paper, you may request it from the proctor.

If you give multiple answers to a question and do not clearly indicate which one you wish to be graded, we reserve the right to select which one to grade. This will likely not be to your advantage. As such, we suggest that you cross out all the scratch work on the question pages that is not part of your answer before you submit the exam.

Please keep this cover page and the question pages intact.

Question	Points	Possible
1		10
2		10
3		10
4		10
Total		40

This space reserved for proctor use

1. (a) Let $S = \{S_1, S_2, \ldots S_n\}$ be a finite collection of finite sets. A *system of distinct representatives*, or SDR, of S is a set of distinct elements $\{x_1, x_2 \ldots x_n\}$ such that-

$$x_1 \in S_1, x_2 \in S_2 \ldots x_n \in S_n$$

.

Give an algorithm to find whether S has an SDR or not. (5 pts)

(b) Consider the class of 3-SAT instances in which each of the n literals (variables) occurs – counting positive and negative appearances combined – in exactly three clauses. Prove that a satisfying assignment for such a problem can always be found in polynomial time. (5 pts)

2. An instance of the *Steiner tree problem* $ST(G, R, k)$ consists of –

   - an undirected graph $G = (V, E)$;
   - a subset of the vertices $R \subseteq V$, called the *terminal nodes*;
   - a number $k \in \mathbb{N}$.

   The $ST(G, R, k)$ problem is "Is there a tree that spans the vertices of R (i.e. a spanning tree for R) and that consists of at most k edges?"

   Prove that the Steiner tree problem is NP-complete. (10 pts)

   Hint: Plant a node in the middle of each edge $e \in E$ of $G(V, E)$. Let this set of new additional nodes be our R. Also connect the nodes V in a complete graph by new edges. Now you can reduce from one of the NP-complete problems we have done in class, to answer that NP-complete problem on $G(V, E)$.

3. (a) Consider a sorted sequence (of *unknown* length) of numbers. Given an efficient algorithm to check whether a specific number x is in the sequence. The time complexity of your algorithm should be logarithmic in the number of elements in the sequence than are less than x. More specifically if the number of elements in the sequence less that x is n, then your algorithm should run in time $O(\log n)$. Note that n differs depending on x and is unknown to the algorithm designer. (5 pts)

   (b) The *element uniqueness problem* is the problem of determining whether all the elements of a list are distinct. The problem may be solved by sorting the list and then checking if there are any consecutive equal elements. The time complexity of such an algorithm would be $O(n \log n)$. Prove that in a model of computation in which only comparison operations on the elements can be performed no algorithm does better than $O(n \log n)$. (5 pts)

4. Suppose that we wish to know which windows in a 36-story building are safe to drop eggs from, and which will cause the eggs to break on landing. We make a few assumptions:

   - An egg that survives a fall can be used again.
   - A broken egg must be discarded.
   - The effect of a fall is the same for all eggs.
   - If an egg breaks when dropped, then it would break if dropped from a higher window.
   - If an egg survives a fall then it would survive a shorter fall.
   - It is not ruled out that the first-floor windows break eggs, nor is it ruled out that the 36th-floor windows do not cause an egg to break.

   Given n eggs and m floors (more generally instead of 36) give an algorithm that finds the minimum number of trials needed to find the window from which it is safe to drop eggs. What is the complexity of your algorithm? (10 pts)

   (Hint: If only one egg is available and we wish to be sure of obtaining the right result, the experiment can be carried out in only one way. Drop the egg from the first-floor window; if it survives, drop it from the second floor window. Continue upward until it breaks. Suppose 2 eggs are available. What is the least number of egg-droppings that is guaranteed to work in all cases?)

Extra page. You may use this for scratch paper, but nothing on this page will be graded.

Extra page. You may use this for scratch paper, but nothing on this page will be graded.