CS180 Final Exam
Winter 2010

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 8:00 AM until 11:00 AM, to work on this exam. If you are still working at 10:40 AM, you will not be permitted to turn in your exam until the end, out of consideration for those who may still be working. When you submit your exam, you may be required to provide a photo ID to the proctor. Exams submitted without the proctor verifying the identity of the test taker may result in the exam being given a zero.

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. 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. Is this problem polynomial time solvable? If so, prove that a satisfying assignment can always be found in polynomial time. If it is not, prove that it is **NP**-Complete.

2. We define the following problem, which we will call 3-SAT(α). We are given a collection of m clauses, each of which contains exactly three literals (variables), and asked to determine whether there is an assignment of true/false values to the literals such that *at least* αm clauses will be true. Note that 3-Sat(1) is exactly the 3-SAT problem.

   (a) Give an $O(mn)$-time algorithm that outputs a satisfying assignment for 3-Sat($\frac{1}{2}$).

   (b) Prove that 3-Sat($\frac{15}{16}$) is **NP**-Complete.
   Hint: Consider the collection of all possible 3-Sat clauses on three literals. Any truth assignment to the literals will make exactly seven of these clauses true (and one false).

3. Graduate students get a lot of free food at various events. Suppose a student has a schedule marked with those days when she gets a free dinner, and those days when she must pay for dinner. On any given day she can buy dinner at the cafeteria for \$6. Alternatively, she can purchase one week's groceries for \$20. However, since she doesn't have a fridge, the groceries will go bad after seven days (including the day of purchase) and any leftovers must be discarded. Give a dynamic programming algorithm to determine, given the schedule of free meals, the minimum amount of money she must spend to make sure she has dinner each night. Prove that your algorithm is correct and has running time polynomial in the number of days on the schedule.

4. We are given n objects $p_1, p_2, \ldots p_n$ and a positive distance between each pair of two objects, such that $d(p_i, p_i) = 0$ and that $d(p_i, p_j) = d(p_j, p_i)$. A k-clustering of these objects is a partitioning into k non-empty sets, and the spacing of a k-clustering is the minimum distance between any pair of objects lying in different clusters. Give an efficient algorithm to find the maximum-spacing k-clustering.

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.