CS 573: Algorithms, Fall 2009

Midterm — October 7, 2009, 7–9 PM

Transportation building, 103

Name:	
Net ID:	Alias:

- This is a closed-book, closed-notes, open-brain exam. If you brought anything with you besides writing instruments and your **handwritten** $8\frac{1}{2}'' \times 11''$ cheat sheet, please leave it at the front of the classroom.
- Print your name, netid, and alias in the boxes above. Print your name at the top of every page (in case the staple falls out!).
- **You should answer all the questions on the exam.**
- The last few pages of this booklet are blank. Use that for a scratch paper. Please let us know if you need more paper.
- If your cheat sheet is not hand written by yourself, or it is photocopied, please do not use it and leave it in front of the classroom.
- Submit your cheat sheet together with your exam. An exam without your cheat sheet attached to it will not be checked.
- If you are NOT using a cheat sheet you should indicate it in large friendly letters on this page.
- There are 4 questions on the exam. Each question is worth 25 points.
- Answers containing *only* the expression: "I dont know", will get 20% of the points of the question. If you write anything else, it would be ignored. Overall, points given for "I dont know" will not exceed 10 points.
- Write your exam using a pen not a pencil.
- Time limit: 120 minutes.
- Relax. Breathe. This is just an easy, silly and stupid midterm.

#	Score	IDK	Grader
1.			
2.			
3.			
4.			
Total			

1. NP COMPLETENESS.
   [25 Points]

   (A) **[5 Points]** Consider the complete graph K_n over n vertices (for n being even) and a maximum cut $(S, V(K_n) \setminus S)$ in K_n. A maximum cut is a cut that has a maximum number of edges in it.

   What is the size of S?

   What is the size of $(S, V(K_n) \setminus S)$?

   Prove your answer. (Hint: You can use the fact that the function $f(x) = x(\alpha - x)$ gets its maximum at $x = \alpha/2$.)

   (B) Prove that the following problem is NP-COMPLETE.

   Problem: Weighted Max Cut

Instance: A graph $G = (V, E)$, a weight function w on the edges of G assigning each edge a non-negative weight, and a number U. *Question:* Is there a cut $(S, V \setminus S)$ in G with total weight $\geq U$?

   The weight of the cut $(S, V \setminus S)$ is $\displaystyle\sum_{v \in S, u \in V \setminus S} w(uv)$. (Hint: Use (A) somehow.)

2. BOUNDED SET COVER.
   [25 Points]

In the set cover problem, you are given an set X of n elements, and a family of m subsets $\mathcal{F}$ of X. The task is to find a minimum number of sets of $\mathcal{F}$ that covers all the elements of X. Let opt denote the number of sets in the optimal solution.

In the bounded set cover problem, you also know that every set has size at most u.

(A) Let α_i be the number of elements covered for the first time in the ith iteration of the greedy algorithm. Prove that $\alpha_i \leq u/2$, for $i > 2\text{opt}$.

(B) Bound the number of sets output by the greedy algorithm as a function of u, n, opt, and m (your function might not depend on all these parameters). Your bound should be as low as possible (ignoring constants). Prove your bound.

3. Not Random walk.
[25 Points]

Let S be a set of n elements $S = \{s_1, \ldots, s_n\}$. At every round, you can randomly pick (with repetition) one element r_i of S (with all elements having probability $1/n$ to be picked). For example, if $S = \{1, 2, 3\}$ it might be that in the first round you picked the number 3, in the second round you picked the number 1, in the third round you picked the number 3 again, and in the fourth round you picked the number 2, and so on.

(A) **[10 Points]** Assume you already picked i distinct elements from S (let I denote this set of distinct elements), and let X_i be the number of additional number of rounds you perform till you pick a new element (i.e., an element that is in $S \setminus I$).

What is the expectation of X_i?

.

(B) **[10 Points]** Starting with the empty set, in expectation, how many rounds one needs to pick all the n distinct elements of X?

(C) **[5 Points]** Let t be the number of rounds one has to perform till one picked all the elements of S with probability larger than $1 - 1/n^{10}$. Give a bound on t, and prove your answer.

4. Interval cover.

[25 Points]

You are given $X = \{1, 2, 3, \ldots, n\}$, and a set $\mathcal{I} = \{I_1, \ldots, I_n\}$ of n intervals on the real line. That is, for every interval I_j you are given two numbers $a_j \in X$ and $b_j \in X$ such that $I_j = [a_j, b_j]$ covers all the numbers of X between a_j and b_j. Describe an algorithm that, in polynomial time, finds a subset of the intervals of $\mathcal{I}$ such that they cover all the numbers of X, and this subset is as small as possible.

What is the running time of your algorithm? Provide a pseudo-code for your algorithm.