CS 573: Algorithms, Fall 2012

Homework 2, due Monday, October 1, 23:59:59, 2012

Version 1.02

Name:	
Net ID:	

Neatly print your name(s), NetID(s). If you are off campus, please submit the homework on moodle, otherwise submit the homework in SC 3306 (or sliding it under the door). Please solve each problem on a separate page. Because of the midterm, this is a slightly smaller homework.

This homework should be easier than hws 0 and 1. You are encouraged to discuss problems in this homework with people, but should submit your homework on your own.

To acknowledge the corn – This purely American expression means to admit the losing of an argument, especially in regard to a detail; to retract; to admit defeat. It is over a hundred years old. Andrew Stewart, a member of Congress, is said to have mentioned it in a speech in 1828. He said that haystacks and cornfields were sent by Indiana, Ohio and Kentucky to Philadelphia and New York. Charles A. Wickliffe, a member from Kentucky questioned the statement by commenting that haystacks and cornfields could not walk. Stewart then pointed out that he did not mean literal haystacks and cornfields, but the horses, mules, and hogs for which the hay and corn were raised. Wickliffe then rose to his feet, and said, "Mr. Speaker, I acknowledge the corn".

– Funk, Earle. A Hog on Ice and Other Curious Expressions.

Required Problems

1. INDEPENDENCE.
   [**40 Points**]
   As you know, computing the largest independent set in a graph is NPC. Here, we are going to investigate an elegant approximation algorithm, that works surprisingly well.
   (A) [**10 Points**] Consider a graph G, and a random permutation of its vertices. Prove that the probability that a vertex v appear in the permutation before all its neighbors is $\frac{1}{1+\mathrm{d}(v)}$.
   (B) [**20 Points**] Consider the algorithm that first randomly permute the vertices of a give graph G, and then scan the vertices one by one. In the ith iteration, it consider the i vertex in the permutation $v_{\pi(i)}$ and add it to the set S if none of its neighbors were added to S. In the end of the execution of the algorithm, clearly the set S is independent. Prove (formally!) that the probability of a vertex v to be in S is at least $\frac{1}{1+\mathrm{d}(v)}$, and the expected size of the independent set being computed is at least $\sum_{v \in \mathsf{V}(\mathsf{G})} \frac{1}{1+\mathrm{d}(v)}$.

(C) **[10 Points]** Prove that the independent set being computed by (B) is, in expectation, of size $\geq \dfrac{|\mathsf{V}(\mathsf{G})|}{1 + \text{average\_degree}(\mathsf{G})}$. (The interested student can verify that this bound is tight in the worst case.)
Hint: Prove that for any positive real numbers $x_1, \ldots, x_n$ with total sum α, it holds that $\sum_{i=1}^{n} \frac{1}{x_i}$ is minimized when $x_1 = x_2 = \cdots = x_n = \alpha/n$. Use this inequality to derive the above claim from (B).

2. Greedy algorithm does not work for coloring. Really.
[30 Points]

Let G be a graph defined over n vertices, and let the vertices be ordered: $v_1, \ldots, v_n$. Let G_i be the induced subgraph of G on $v_1, \ldots, v_i$. Formally, $\mathsf{G}_i = (\mathsf{V}_i, E_i)$, where $\mathsf{V}_i = \{v_1, \ldots, v_i\}$ and

$$\mathsf{E}_i = \left\{ uv \in \mathsf{E} \;\middle|\; u, v \in \mathsf{V}_i \text{ and } uv \in \mathsf{E}(G) \right\}.$$

The greedy coloring algorithm, colors the vertices, one by one, according to their ordering. Let k_i denote the number of colors the algorithm uses to color the first i vertices.

In the ith iteration, the algorithm considers v_i in the graph G_i. If all the neighbors of v_i in G_i are using all the k_{i-1} colors used to color G_{i-1}, the algorithm introduces a new color (i.e., $k_i = k_{i-1} + 1$) and assigns it to v_i. Otherwise, it assign v_i one of the colors $1, \ldots, k_{i-1}$ (i.e., $k_i = k_{i-1}$).

Give an example of a graph G with n vertices, and an ordering of its vertices, such that even if G can be colored using $O(1)$ (in fact, it is possible to do this with two) colors, the greedy algorithm would color it with $\Omega(n)$ colors. (Hint: consider an ordering where the first two vertices are not connected.)

3. Maximum Clique
[30 Points]

Let $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ be an undirected graph. For any $k \geq 1$, define $\mathsf{G}^{(k)}$ to be the undirected graph $\left(\mathsf{V}^{(k)}, \mathsf{E}^{(k)}\right)$, where $\mathsf{V}^{(k)}$ is the set of all ordered k-tuples of vertices from V and $\mathsf{E}^{(k)}$ is defined so that $(v_1, v_2, ..., v_k)$ is adjacent to $(w_1, w_2, ..., w_k)$ if and only if for each i (for $i = 1, \ldots, k$) either vertex v_i is adjacent to w_i in G, or else $v_i = w_i$.

(A) **[10 Points]** Prove that the size of the maximum clique in $\mathsf{G}^{(k)}$ is equal to the kth power of the size of the maximum clique in G. That is, if the largest clique in G has size α, then the largest clique in $\mathsf{G}^{(k)}$ is α^k, and vice versa.

(B) **[10 Points]** Show an algorithm that is given a clique of size β in $\mathsf{G}^{(k)}$ and outputs a clique of size $\left\lceil \beta^{1/k} \right\rceil$ in G.

(C) **[5 Points]** Argue that if there is an c-approximation algorithm for maximum clique (i.e., it returns in polynomial time a clique of size $\geq \text{opt}/c$) then there is a polynomial time $c^{1/k}$-approximation algorithm for maximum clique, for any constant k. What is the running time of your algorithm, if the running time of the original algorithm is $T(n)$. (Hint: use (A) and (B).)

(D) **[5 Points]** Prove that if there is a constant approximation algorithm for finding a maximum-size clique, then there is a polynomial time approximation scheme for the problem.[1]

[1] Can one prove that there is FPTAS in this case? I do not know.