Assignment #6 CS4/531

Due Date: Friday, Dec. 10, 2010

UNSUPPORTED SOLUTIONS RECEIVE NO CREDIT.

Total points: 43

- You MUST turn in your HW by 4:10pm on Dec. 10. After that, I will NOT accept your HW. This rule will be **STRICTLY ENFORCED.**
- Please PRINT YOUR LAST NAME, FIRST NAME and UB number on the first page.
- Write solution of each problem on a separate sheet. Staple them in the order of problem numbers.
- If your homework solution deviates significantly from these guidelines, TA may deduct up to 20% of the points.

1. (7 points). A film producer is seeking actors and investors for his new movie. There are n available actors $A = \{a_1, a_2, \dots, a_n\}$. Actor a_i $(1 \le i \le n)$ charges s_i dollars. For the funding, there are m available investors $I = \{b_1, b_2, \dots, b_m\}$. Investor b_j $(1 \le j \le m)$ will provide p_j dollars, but only on the condition that certain actors $L_j \subseteq A$ are included in the cast. (**All** actors in L_j must be included, otherwise the investor b_j **will invest nothing**. Talking about big Hollywood ego!)

The financial arrangement is that all income from the film (such as movie tickets sales, DVD sales etc) belong to the investors. (So the investors' profit/loss depends on the income of the film). Thus the producer's profit will be the sum of the payments from investors minus the payments to actors. The goal of the producer is, of course, to maximize his profit.

Describe an algorithm that finds the maximum profit for the producer.

Note: This is an example that seemingly has nothing to do with the Max-flow problem, yet can be converted to a max-flow problem. Consider the following flow network G. G has a source s, the vertices in A, the vertices in I, and a sink t. For each $i \in \{1, 2, \dots, n\}$, there is an edge $s \to a_i$ of capacity s_i. For each $j \in \{1, 2, \dots, m\}$, there is an edge $b_j \to t$ of capacity p_i. For each $i \in \{1, 2, \dots, n\}$ and each $j \in \{1, 2, \dots, m\}$, if $a_i \in L_j$ then there is an edge $a_i \to b_j$ of infinite capacity.

(a) Let (S, T) be any cut of G with finite capacity. Show that if $b_j \in T$ then $a_i \in T$ for all $a_i \in L_j$.

(b) Show how to determine the maximum profit from the capacity of the minimum cut of G and the given s_i and p_j values.

2. (7 points). Minimum Vertex Cover for Bipartite Graph (MVCBG) Problem.

Let $G = (V, E)$ be an undirected graph. A *vertex cover* (VC) of G is a subset of vertices $C \subseteq V$ such that for any edge $e = (u, v) \in E$, at least one end vertex of e (either u or v or both) is in C. A VC C of G is a *minimum vertex cover* (MVC) if its size $|C|$ is minimum among all vertex covers of G. The MVC problem is: Given input graph G, find a MVC of G.

We mentioned in class that MVC is NP-complete, hence it is highly unlikely the problem can be solved in polynomial time.

However, if we restrict the input to be bipartite graphs, the problem is easier:

Minimum Vertex Cover for Bipartite Graph (MVCBG) Problem:

Input: A bipartite undirected graph $G = (X, Y, E)$.

Output: A MVC of G.

Note: This is yet another example that is seemingly unrelated to max-flow, but can be converted to a max-flow problem. Construct a flow network $\bar{G}$ as follows. $\bar{G}$ has a source s, a sink t and all vertices in $X \cup Y$. There is an edge $s \to x$ for every $x \in X$ with capacity 1 and an edge $y \to t$ for every $y \to t$ with capacity 1. For every edge $(x, y) \in E$ (where $x \in X$ and $y \in Y$), there is an edge $x \to y$ in $\bar{G}$ with capacity 2.

Hint: Try to find a connection between a minimum cut (S, T) of $\bar{G}$ and a MVC C of G. Think it along these lines. Suppose that (S, T) is a cut of $\bar{G}$. Define $X_S = X \cap S$, $X_T = X \cap T$ and $Y_S = Y \cap S$, $Y_T = Y \cap T$. How do you get a VC of G from X_S, X_T, Y_S, Y_T? Conversely, suppose C is a VC of G. Define $X_C = X \cap C$ and $Y_C = Y \cap C$. How do you get a cut of $\bar{G}$ from X_C and Y_C?

3. (3+3+3=9 points). The **Graph Coloring** (GC) problem is defined as follows. The input is an undirected graph $G = (V, E)$ with n vertices and m edges. A *vertex coloring* of G is an assignment of colors to the vertices of G so that for any two vertices u, v, if (u, v) is an edge of G, u and v are assigned different colors. The problem is: Given G, determine the smallest integer k such that G has a vertex coloring using k colors. (Note that G can be colored by two colors iff G is bipartite.)

(a) Describe the decision version of the GC problem.

(b) Suppose that we have an algorithm A for solving the decision version of the GC problem with run time $T(n, m)$. Describe an algorithm B for solving the optimization version of the GC problem. Analyze the run time of B (in terms of $T(n, m)$.)

(c) Consider the following **Task Scheduling** (TS) problem. There are unlimited number of processors at a computing center. There are n *tasks* $T_1, T_2, \ldots, T_n$ to be processed on these processors. Each task can be processed by any processor in one time slot. When a task T_i is being processed, it needs to access certain files. If two tasks T_i and T_j need to access the same file, then T_i and T_j cannot be processed during the same time slot, due to file access conflicts. The problem is: How to schedule the tasks so that all tasks are processed within minimum number of time slots.

Describe how to convert the TS problem to the GC problem.

4. (6 points) Let $G = (V, E)$ be an undirected graph. A **Hamiltonian Path** (HP) of G is a path that visits each vertex of G exactly once. A **Hamiltonian Cycle** (HC) of G is a cycle that visits each vertex of G exactly once.

The *Hamiltonian Cycle* (HC) problem is: Given an undirected graph G, decide if G has a Hamiltonian Cycle or not. The *Hamiltonian Path* (HP) problem is: Given an undirected graph G, decide if G has a Hamiltonian path or not.

Show that HP is polynomial time reducible to HC. Namely show HP $\leq_P$ HC.

5. (3+5= 8 points) Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs. We say G_1 *and* G_2 *are isomorphic* if there is a one-to-one mapping $f : V_1 \to V_2$ such that for any two vertices $u, v \in V_1$, $(u, v) \in E_1$ if and only if $(f(u), f(v)) \in E_2$. The two graphs G_1, G_2 shown in Figure 1 are isomorphic. (As you can see, if two graph G_1 and G_2 are isomorphic, then they are really the same graph with different vertex names).

Sub-graph Isomorphic (SGI) problem:

Input: Two undirected graphs G and K.

Output: Yes, if K has a subgraph H and H is isomorphic to G; no. otherwise.

For example, If we take $G = G_1$ and $K = G_3$ in Figure 1 as the input, the answer should be yes. (Because $H = G_2$ is a subgraph of $K = G_3$ and $H = G_2$ is isomorphic to $G = G_1$.)

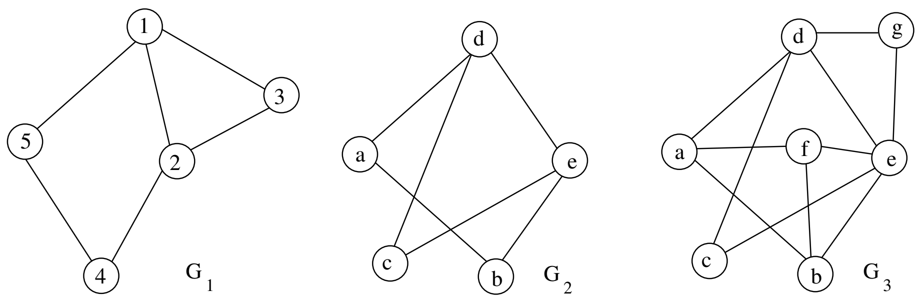

Figure 1: G_1 ad G_2 are isomorphic, by the mapping: $f(1) = d, f(2) = e, f(3) = c, f(4) = b, and f(5) = a$.

1. Prove this problem is in NP. This can be done by describing either a polynomial time verification algorithm (with a polynomial size certificate); or by a polynomial time non-determinate algorithm for solving it.

2. Prove this problem is NP-complete. This can be done by picking a known NPC problem Q; and show $Q \leq_{\mathcal{P}} SGI$.

6. (6 points) Page 1111, Problem 35.1-4.

This is the minimum vertex cover (MVC) problem for trees. We know that the MVC problem for general graphs is NPC. However, if the input graph is restrict to trees, it is much easier.