HW 9 (due Monday, at noon, April 15, 2013)

CS 473: Fundamental Algorithms, Spring 2013 Version: **1.01**

Make sure that you write the solutions for the problems on separate sheets of paper. Write your name and netid on each sheet.

Collaboration Policy: The homework can be worked in groups of up to 3 students each.

---

1. (50 PTS.) We want a proof!

The following question is long, but not very hard, and is intended to make sure you understand the following problems, and the basic concepts needed for proving NP-Completeness.

For each of the following problems, you are given an instance of the problem of size n. Imagine that the answer to this given instance is "yes". Imagine that you need to convince somebody that indeed the answer to the given instance is yes – to this end, describe:

(I) The format of the proof that the instance is correct.

(II) A bound on the length of the proof (its have to be of polynomial length in the input size).

(III) An efficient algorithm (as fast as possible [it has to be polynomial tie]) for verifying, given the instance and the proof, that indeed the given instance is indeed positive.

We solve the first such question, so that you understand what we want.[1]

(A) (0 PTS.)

Shortest Path

> **Instance**: A weighted undirected graph G, vertices s and t and a threshold w.
> **Question:** Is there a path between s and t in G of length at most w?

Solution: A "proof" in this case would be a path π in G (i.e., a sequence of at most n vertices) connecting s to t, such that its total weight is at most w. The algorithm to verify this solution, would verify that all the edges in the path are indeed in the graph, the path starts at s and ends at t, and that the total weight of the edges of the path is at most w. The proof has length $O(n)$ in this case, and the verification algorithm runs in $O(n^2)$ time. if we assume graph is given to us using an adjacency lists.

(B) (5 PTS.)

Independent Set

> **Instance**: A graph G, integer k
> **Question:** Is there an independent set in G of size k?

---

[1] We trust that the reader can by now readily translate all the following questions to questions about climbing vampires from Champaign. The reader can do this translation in their spare time for their own amusement.

(C) (5 PTS.)

3Colorable

Instance: A graph G.
Question: Is there a coloring of G using three colors?

(D) (5 PTS.)

TSP

Instance: A weighted undirected graph G, and a threshold w.
Question: Is there a TSP tour of G of weight at most w?

(E) (5 PTS.)

Vertex Cover

Instance: A graph G, integer k
Question: Is there a vertex cover in G of size k?

(F) (5 PTS.)

Subset Sum

Instance: S - set of positive integers,t: - an integer number (target).
Question: Is there a subset $X \subseteq S$ such that $\sum_{x \in X} x = t$?

(G) (5 PTS.)

3DM

Instance: X, Y, Z sets of n elements, and T a set of triples, such that $(a, b, c) \in T \subseteq X \times Y \times Z$.
Question: Is there a subset $S \subseteq T$ of n disjoint triples, s.t. every element of $X \cup Y \cup Z$ is covered exactly once.?

(H) (5 PTS.)

Partition

Instance: A set S of n numbers.
Question: Is there a subset $T \subseteq S$ s.t. $\sum_{t \in T} t = \sum_{s \in S \setminus T} s$?

(I) (5 PTS.)

SET COVER

Instance: $(X, \mathcal{F}, k)$:
X: A set of n elements
$\mathcal{F}$: A family of subsets of S, s.t. $\bigcup_{X \in \mathcal{F}} X = X$.
k: A positive integer.
Question: Are there k sets $S_1, \ldots, S_k \in \mathcal{F}$ that cover S. Formally, $\bigcup_i S_i = X$?

(J) (5 PTS.)

CYCLE HATER.

> **Instance**: An undirected graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$, and an integer $k > 0$.
> **Question:** Is there a subset $X \subseteq \mathsf{V}$ of at most k vertices, such that all cycles in G contain at least one vertices of X.

(K) (5 PTS.)

CYCLE LOVER.

> **Instance**: An undirected graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$, and an integer $k > 0$.
> **Question:** Is there a subset $X \subseteq \mathsf{V}$ of at most k vertices, such that all cycles in G contain at least two vertices of X.

2. (50 PTS.) Beware of Greeks bearing gifts.[2]
The woodland deity, the brother of the ***induction fairy***, came to visit you on labor day, and left you with two black boxes.

(A) (25 PTS.) The first black-box can solve Partition in polynomial time (note that this black box solves the decision problem). Let S be a given set of n integer numbers. Describe a polynomial time algorithm that computes, using the black box, a partition of S if such a solution exists. Namely, your algorithm should output a subset $T \subseteq S$, such that

$$\sum_{s \in T} s = \sum_{s \in S \setminus T} s.$$

(B) (25 PTS.) The first box was a black box for solving Subgraph Isomorphism.

Subgraph Isomorphism

> **Instance**: Two graphs, $G = (V_1, E_1)$ and $H = (V_2, E_2)$.
> **Question:** Does G contain a subgraph *isomorphic* to H, that is, a subset $V \subseteq V_1$ and a subset $E \subseteq E_1$ such that $|V| = |V_2|$, $|e| = |E_2|$, and there exists a one-to-one function $f : V_2 \to V$ satisfying $\{u, v\} \in E_2$ if and only if $\{f(u), f(v)\} \in E$?

Show how to use this black box, to compute the subgraph isomorphism (i.e., you are given G and H, and you have to output f) in polynomial time. (You can assume that a call to this black box, takes polynomial time.)

[2]The expression "beware of Greeks bearing gifts" is based on Virgil's Aeneid: "Quidquid id est, timeo Danaos et dona ferentes", which means literally "Whatever it is, I fear Greeks even when they bring gifts."