HW 10 (due Monday, Noon, April 27, 2015)

OLD CS 473: Fundamental Algorithms, Spring 2015 Version: **1.02**

Collaboration Policy: For this homework, Problems 1–3 can be worked in groups of up to three students.

1. (30 pts.) Many far cycles.

Consider the following problem.

Many cycles

> **Instance**: An undirected graph G, vertices s and t and threshold k and ℓ (both positive integer numbers.
> **Question:** Are there k vertex disjoint simple cycles $C_1, \ldots, C_k$ in G, where each cycle has most ℓ vertices, and furthermore, there is no edge in G between any two vertices that belong to different cycles in $C_1, \ldots, C_k$.

Either prove that this problem is NP-Complete, or provide a polynomial time algorithm for solving it.

2. (30 pts.) Many ℓ-spiders.

Either prove that the following problem is NP-Complete, or provide a polynomial time algorithm for solving it.

Many ℓ-Spiders.

> **Instance**: An undirected graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ an integer k, and an integer ℓ.
> **Question:** Are there k vertex-disjoint spiders, each with ℓ legs, that visits all the vertices of G?

A ***spider*** with ℓ legs in a graph G is defined by a vertex v (i.e., the head of the spider), and a collection Π of ℓ simple paths all starting in v, that except for v, are vertex disjoint. The vertex set of such a spider is all the vertices that the paths of Π visit.

3. (40 pts.) Beware of Greeks bearing gifts

(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.".) The reduction faun, the adopted uncle of the Partition satyr, came to visit you on unofficial, and left you with four green gifts. One of them was 2,000 euros in Greek government bonds – now almost worthless. The other three gifts are black boxes that can solve decision problems. Show how to use these black boxes:

(A) (10 pts.) The first black box can solve the following decision problem in polynomial time:

Many Spiders.

> **Instance**: An undirected graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ and an integer k.
> **Question:** Are there k vertex-disjoint spiders that visits all the vertices of G?

Show how to use this black box to solve, in polynomial time, the optimization version of this problem (i.e., finding the minimum number of spiders needed to cover all the vertices in a graph, and outputting these spiders).

(B) (10 pts.) The next 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

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

(C) (20 pts.) The final black box can solve the following decision problem.

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 simple cycles in G contain at least two vertices of X.

Given a graph G describe a polynomial time algorithm, that uses the above black box, that computes the smallest set $X \subseteq \mathsf{V}(\mathsf{G})$, such that every simple cycle in G contains at least two vertices of X. What is the running time of your algorithm, ignoring the calls to the black box?