CS 473: Fundamental Algorithms, Spring 2011

HW 10

Question 1 is due by **Sunday, 23:59:59, April 24**

Questions 2-4 are due by **Monday, 23:59:59, April 25**

This homework contains four problems. **Read the instructions for submitting homework on the course webpage**.

Collaboration Policy: For this homework, Problems 2–4 can be worked in groups of up to three students.

Problem 1 should be answered in Compass as part of the assessment HW10-Online and should be done individually.

---

1. HW10-Online. (25 pts.)

2. All we are saying is give flow a chance. (25 pts.)

   You are given a flow network G, with source s and sink t. All the edges have capacity 1.

   (A) (10 pts.) You need to decide if there is an integral flow from s to t of value 1, such that: (i) there is no cycle in the flow, and (ii) the flow goes through all the vertices in the graph.
   Does this problem has a polynomial time algorithm? Or is it NP-Complete? If it has a polynomial time algorithm, then show the algorithm. If it is NP-Complete, then prove that it is NP-Complete.

   (B) (15 pts.) Consider the variant of the above problem where some edges are marked, and you need to find a unit flow that is acyclic and uses all the marked edges. Is this problem NP-Complete? Is it solvable in polynomial time? Prove your answer.

3. A visit. (25 pts.)

   (A) (10 pts.) The Partition satyr, the uncle of the deduction fairy, had visited you on winter break and gave you, as a token of appreciation, a black-box that 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

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

   In the Partition problem, an instance is a set of n numbers N, and the question is whether one can find a subset $X \subseteq N$ such that $\sum_{x \in X} x = \sum_{x \in N \setminus X} x$.

(B) (15 pts.)

Show that an algorithm that makes at most a constant number of calls to polynomial-time subroutines runs in polynomial time, but that a polynomial number of calls to a polynomial-time subroutines may result in an exponential-time algorithm.

4. UNICORNS AND COLORING. (25 pts.)

The invisible pink unicorn (see Wikipedia for a picture), the sister of the PARTITION satyr, had visited you on spring break and gave you, as a token of appreciation, a black-box that can solve 15COLORING in polynomial time (note that this black box solves the decision problem. That is, given an undirected graph the black-box can tell you in polynomial time whether or not the graph can be colored with 15 colors such that no edge has its both vertices colored by the same color).

(A) (10 pts.) Show how to build a procedure that decides given a graph if it is 4-colorable in polynomial time (using the given block box).

(B) (15 pts.) Describe a polynomial time algorithm that its input is an undirected graph, and using the above black box, it computes a coloring of the given graph by three colors if such a coloring exists, and if not declares that such a coloring is impossible.