CS 5633 Analysis of Algorithms – Spring 12

4/19/12

11. Homework

Due **4/26/12** before class

1. **NP-completeness**

   (a) The **Zero-Sum** problem takes a set A of n integers as input, and asks whether there is a non-empty subset of A that sums to zero. The **Subset-Sum** problem takes as input a set A of n integers and a number S, and it asks whether there is a non-empty subset of A that sums to S.

   Assume you know that **Subset-Sum** is NP-complete. Prove that **Zero-Sum** is NP-complete.

   (b) The **2-TSP** problem takes an undirected graph $G = (V, E)$ with positive edge weights as well as a positive integer k as input, and asks whether there are **two** closed tours in G such that each tour contains at least two vertices, both tours together visit every vertex in V exactly once, and the total sum of all edge weights on both tours is at most k. Prove that **2-TSP** is NP-complete.

2. $\Pi_1 \leq \Pi_2$

   Let Π_1 and Π_2 be decision problems and suppose Π_1 is polynomial-time reducible to Π_2, so, $\Pi_1 \leq \Pi_2$. Answer and justify each of the questions below:

   (a) If $\Pi_1 \in NP$, does this imply that Π_2 is NP-complete?

   (b) If $\Pi_1 \in NP$, does this imply that $\Pi_2 \in NP$?

   (c) If $\Pi_2 \in NP$, does this imply that Π_1 is NP-complete?

   (d) If $\Pi_2 \in NP$, does this imply that $\Pi_1 \in NP$?

   (e) If $\Pi_1 \notin P$ does this imply that $\Pi_2 \notin P$?

   (f) If $\Pi_2 \notin P$ does this imply that $\Pi_1 \notin P$?

   (g) If Π_2 is NP-complete, what implications does that have for Π_1?

   (h) If Π_1 is NP-complete, what implications does that have for Π_2?

   (i) If $\Pi_1 \in NP$ and $\Pi_2 \in P$, what does this imply?

   (j) If Π_1 is NP-complete and there are an exponential-time algorithm for Π_2 as well as a polynomial-time algorithm for Π_2, what does this imply?

3. **Vertex Cover for trees**

   Develop a linear-time greedy algorithm that finds an optimal vertex cover for a tree. Argue why the cover computed by your algorithm is indeed an optimal vertex cover.

4. **Euclidean TSP**

   Consider the variant of the TSP problem in which the vertices are points in the plane and the cost of an edge between two vertices is their Euclidean distance. Show that an optimal TSP tour never intersects itself.