CS612 Assignment 5

Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China

November 27, 2009

Notice:

1. Due Dec. 17, 2009.
2. Please send your answer to wangchao1987@ict.ac.cn, shaomingfu@gmail.com, yuanxiongying@ict.ac.cn
3. You can arbitrarily choose two problems from Problems 1-5.

1 Greedy Algorithm(5 marks)

Let us say that a graph $G = (V,\ E)$ is a $near - tree$ if it is connected and has at most $n + 8$ edges, where $n = |V|$. Give an algorithm with running time $O(n)$ that takes a near-tree G with costs on its edges, and returns a minimum spanning tree of G. You may assume that all the edge costs are distinct.

2 Greedy Algorithm(10 marks)

Given a list of n natural numbers d_1, d_2,...,d_n, show how to decide in polynomial time whether there exits an undirected graph $G = (V,\ E)$ whose node degrees are precisely the numbers d_1, d_2,...,d_n. G should not contain multiple edges between the same pair of nodes, or " loop" edges with both endpoints equal to the same node.

3 Greedy Algorithm(10 marks)

Suppose you are given a directed graph $G = (V,\ E)$ in which each edge has a cost of either 0 or 1. Also suppose that G has a node r such that there is a path from r to every other node in G. You are also given an integer k. Give a polynomial-time algorithm that either constructs an arborescence rooted at r of cost **exactly** k, or reports that no such arborescence exists.

4 Greedy Algorithm(10 marks)

The input consists of n skiers with heights p_1, p_2,...,p_n, and n skies with height s_1, s_2,...,s_n. The problem is to assign each skier a ski to minimize the **AVERAGE DIFFERENCE** between the height of a skier and his/her assigned ski. That is, if the skier i is given the ski a_i, then you want to minimize:

$$\Sigma_{i=1}^{n}(|p_i - s_{a_i}|)/n$$

5 Greedy Algorithm(10 marks)

The input to this problem consists of an ordered list of n words. The length of the ith word is w_i, that is the ith word takes up w_i spaces. The goal is to break this ordered list of words into lines, this is called a layout. Note that you can not reorder the words. The length of a line is the sum of lengths of the words on that line. The ideal line length is L. No line may be longer than L, although it may be shorter. The penalty of having a line of length K is $L-K$. The total penalty is the **sum** of the line penalties. The problem is to find the layout that minimizes the total penalty.