

CS 573: Graduate Algorithms, Fall 2010

Homework 3

Due Monday, October 18, 2010 at 5pm
(in the homework drop boxes in the basement of Siebel)

1. Suppose we are given two arrays $C[1..n]$ and $R[1..n]$ of positive integers. An $n \times n$ matrix of 0s and 1s *agrees with R and C* if, for every index i , the i th row contains $R[i]$ 1s, and the i th column contains $C[i]$ 1s. Describe and analyze an algorithm that either constructs a matrix that agrees with R and C , or correctly reports that no such matrix exists.
2. Suppose we have n skiers with heights given in an array $P[1..n]$, and n skis with heights given in an array $S[1..n]$. Describe an efficient algorithm to assign a ski to each skier, so that the average difference between the height of a skier and her assigned ski is as small as possible. The algorithm should compute a permutation σ such that the expression

$$\frac{1}{n} \sum_{i=1}^n |P[i] - S[\sigma(i)]|$$

is as small as possible.

3. Alice wants to throw a party and she is trying to decide who to invite. She has n people to choose from, and she knows which pairs of these people know each other. She wants to pick as many people as possible, subject to two constraints:
 - For each guest, there should be at least five other guests that they already know.
 - For each guest, there should be at least five other guests that they *don't* already know.

Describe and analyze an algorithm that computes the largest possible number of guests Alice can invite, given a list of n people and the list of pairs who know each other.

4. Consider the following heuristic for constructing a vertex cover of a connected graph G : return the set of non-leaf nodes in any depth-first spanning tree of G .
 - (a) Prove that this heuristic returns a vertex cover of G .
 - (b) Prove that this heuristic returns a 2-approximation to the minimum vertex cover of G .
 - (c) Describe an infinite family of graphs for which this heuristic returns a vertex cover of size $2 \cdot OPT$.

5. Suppose we want to route a set of N calls on a telecommunications network that consist of a cycle on n nodes, indexed in order from 0 to $n - 1$. Each call has a source node and a destination node, and can be routed either clockwise or counterclockwise around the cycle. Our goal is to route the calls so as to minimize the overall load on the network. The load L_i on any edge $(i, (i + 1) \bmod n)$ is the number of calls routed through that edge, and the overall load is $\max_i L_i$. Describe and analyze an efficient 2-approximation algorithm for this problem.