CS 170 Efficient Algorithms and Intractable Problems
Spring 2017 Prasad Raghavendra and Sanjam Garg

Midterm 2 Review

March 18, 2017

1. Short Questions

1. Give a Huffman encoding tree for the alphabet $\{a,b,c,d,e\}$ with frequencies $f_a = 0.15$, $f_b = 0.32$, $f_c = 0.19$, $f_d = 0.24$, $f_e = .1$.

2. Consider an undirected graph $G = (V,E)$ with nonnegative edge weights $w_e \geq 0$. Suppose that you have computed a minimum spanning tree of G, and that you have also computed shortest paths to all nodes from a particular node $s \in V$.

   Now suppose each edge weight is increased by 1: the new weights are $w'_e = w_e + 1$.

   (a) Assuming G has a unique minimum spanning tree T, is T still an MST of the new graph? Prove or provide a counterexample.

   (b) Can the shortest path between two vertices change? Prove that it can't or provide an example in which it does.

3. Under a Huffman encoding of n symbols with frequencies $f_1, f_2, \ldots, f_n$, what is the longest a codeword could possibly be? Give an example set of frequencies that would produce this case.

4. We have an MST $T = (V,E')$ of a graph $G = (V,E)$. If we increase the weight of some $e \in E'$, give an algorithm to determine the new MST T'.

2. Lexicographically smallest subsequence

You are given a string S and a length k. Give a dynamic programming algorithm to find the smallest (by lexicographical ordering) subsequence of length exactly k. Note that a subsequence must retain characters in the same ordering as in S, but need not be contiguous. For example: in the word "rocket", the smallest subsequence of length 3 is "cet". For the purposes of this question you can assume concatenating two strings together takes constant time.

3. Longest common increasing subsequence

You are given two sequences $a = a_1, a_2, \ldots, a_n$, and $b = b_1, b_2, \ldots, b_m$, and you want to find a subsequence such that it is a subsequence of both a and b, is strictly increasing, and is as long as possible.

For example, the longest common increasing subsequence of $(1,2,4,2,1,6,8)$ and $(2,1,4,2,4,4,2,6)$ is $(1,2,4,6)$. We can verify that there is no longer subsequence that is both strictly increasing and a subsequence both sequences.

Devise an efficient algorithm to solve return the length of the longest common increasing subsequence (algorithms can optionally find and return the subsequence as well).

Input: Two sequences of integers a, b.

Output: A single integer, denoting the length of the longest common increasing subsequence.

The runtime of your algorithm should be $O(nm(n+m))$.

4. Disease prevention

You are in charge of preventing the spread of a disease in a faraway nation called Treeland. The cities in Treeland are labeled from 1 to n. In addition, some cities are considered neighbors, and you are given that as a list of pairs. Interestingly, the cities are connected in such a way that they form the nodes of a connected undirected, unweighted tree. You'd like to open some disease prevention clinics in some of the cities in such a way that all cities become "safe". A city is "safe" if either (a) it has a disease prevention clinic, or (b) **all** of its neighbors have a disease prevention clinic. The cost of opening a disease prevention clinic in city i is equal to c_i. Each c_i will be positive. Given n, the number of cities, the list of pairs of cities that are considered neighbors, and $c_1, c_2, \ldots, c_n$, devise an efficient algorithm to determine the minimum cost you need to spend in order to make every city safe. Your algorithm may optionally return the set of cities in which we should open a clinic.

Input: An integer n, and a list of $n-1$ pairs of integers, where each pair describes two neighboring cities. In addition, you are given a length n array of positive integers $c_1, \ldots, c_n$.

Output: A single integer, denoting the minimum cost to make every city safe.

The runtime of your algorithm should be at most $O(n)$.

5. Water Supply via Linear Programming

There are four major cities and three water reservoirs in California.

- Reservoir i holds G_i gallons of water, while city j needs D_j gallons of water.
- It costs p_{ij} dollars per gallon of water supplied from reservoir i to city j.
- The capacity of the piping from reservoir i and city j can handle at most c_{ij} gallons.
- In view of fairness, no city must get more than $1/3$rd of all its water demand from any single reservoir.

Write a linear program to determine how to supply the water from the reservoirs to the cities, at the lowest cost.

1. What are the variables of the linear program, and what do they indicate?

2. What is the objective function being maximized/minimized?

3. What are the constraints of your linear program? (No need to list every constraint, but list one of each type and explain how the rest are generated)