CSE421: Design and Analysis of Algorithms	Feb 14, 2018
Homework 6	
Shayan Oveis Gharan	Due: Feb 22, 2018 at 5:00 PM

Please see https://courses.cs.washington.edu/courses/cse421/18wi/grading.html for general guidelines about Homework problems.

Most of the problems only require one or two key ideas for their solution. It will help you a lot to spell out these main ideas so that you can get most of the credit for a problem even if you err on the finer details. Please justify all answers.

P1) Here we shall show how to use MSTs to get an approximation algorithm to the famous traveling salesman problem. You are given an undirected graph, where every edge $\{u, v\}$ has a non-negative weight $c_{u,v}$. The goal is to find the lowest weight tour of the graph. A tour is a sequence of vertices $v_1, \dots, v_r$ such that $v_1 = v_r$, every vertex of the graph is visited *at least* once, and for every i, $\{v_i, v_{i+1}\}$ is an edge of the graph. The cost of the tour is

$$\sum_{i=1}^{r-1} c_{v_i,v_{i+1}}.$$

Give a polynomial time algorithm to find a tour that is at most twice the optimum tour, i.e., you want to design a polynomial time algorithm with approximation ratio 2.

a) Prove that the weight of the optimum tour is at least the weight of the MST.

b) Perform depth-first search on an MST of the graph to obtain a tour. Write down the code and show that it outputs a tour.

c) Show that the cost of the tour that you find in the previous part is twice the cost of the MST.

P2) Draw the dynamic programming table of the following instance of the knapsack problem: You are 5 items with weight $1, 3, 6, 7, 9$ and value $1, 5, 12, 18, 25$ respectively and the size of your knapsack is 13..

P3) A sequence of numbers $a_1, \dots, a_m$ is called *mirrored* if m is even and for all $1 \leq i \leq m$, $a_i = -a_{m-i+1}$. For example, $3, -4, 4, -3$ is a mirrored sequence. Given a sequence of numbers $x_1, \dots, x_n$ design an $O(n^2)$ time algorithm that outputs the length of the longest mirrored subsequence. For example, on input $1, 5, -1, 3$ the longest mirrored subsequence is $1, -1$ and its length is 2.

HINT: For $i < j$, let $p(i, j)$ denote the length of the longest mirrored subsequence in $x_i, \dots, x_j$. Express $p(i, j)$ in terms of $p(i + 1, j), p(i, j - 1), p(i + 1, j - 1)$. Evaluate the values p(i, j) in order of increasing $|i - j|$.

P4) Suppose we have a path with $n + 1$ vertices numbered $0, \dots, n$. We want to take a package from vertex 0 to vertex n. There are m mailmen on this line, where the i-th mailman is located at $p[i]$, i.e., array p has the location of all mailmen. For each mailman, i, let $v[i]$ be the speed of i, e.g., if $v[i] = 3$ it means that mailman i goes from vertex a to $a + 1$ or $a - 1$ in 1/3 of a second. To move the package, a mailman should pick it up at point 0 and move to a vertex a_1, at that point another mailman can move the package to a vertex a_2 and so on until the package reaches point n. The goal is to minimize the time that it takes to take the package to vertex n. You can assume all entries of p, v are integers. We want to design a polynomial time algorithm that outputs the minimum number of seconds needed to do this job.

Here is an example: Suppose $n = 10, m = 2, p[1] = 2, p[2] = 6, v[1] = 1, v[2] = 2$. One strategy is that the second mailman goes to 0 (in 3 seconds) picks up the package and goes to n (in 5 seconds). This would take 8 seconds. But, the optimum strategy is that mailman 1 goes to 0 (in 2 seconds) picks up the package and goes to 1 (in 1 second) meanwhile mailman 2 goes to 1 (because it takes him only 2.5 seconds to go to 1. He grabs the package and takes it to n in 4.5 seconds. So by this strategy the package will reach n in $2 + 1 + 4.5 = 7.5$ seconds.

a) Prove that if in the optimum solution, the package is handed from mailman i to mailman j at some point, we must have $v[j] > v[i]$.

b) Design a polynomial time algorithm that outputs the minimum number of seconds needed to do the job.

   Hint: Sort the mailmen based on their speed and assume $v[1] \leq v[2] \leq \ldots v[m]$. For all i, j let $p(i, j)$ be the minimum number of seconds needed to pick up the package from 0 and move it to vertex j using only mailmen $1, \ldots, i$.

P5) **Extra Credit:** Given a sequence of positive numbers $x_1, \ldots, x_n$ and an integer k, design a polynomial time algorithm that outputs

$$\sum_{S \in \binom{n}{k}} \prod_{i \in S} x_i,$$

where the sum is over all subsets of size k.