COMP523 Tutorial 4

Coordinator: Aris Filos-Ratsikas Demonstrator: Michail Theofilatos

October 15, 2019

Problem 1

Consider the *fractional knapsack problem*, in which there is a set of n *infinitely divisible* items with values v_i, for $i = 1, \ldots, n$ and weights w_i, for $i = 1, \ldots, n$, and there is a total weight constraint W. The goal is to find fractions $(x_1, \ldots, x_n)$ of each item, with $0 \leq x_i \leq 1$ such that $\sum_{i=1}^{n} x_i \cdot v_i$ is maximised, subject to the total weight constraint $\sum_{i=1}^{n} x_i \cdot w_i \leq W$.

Design an optimal polynomial time greedy algorithm for the fractional knapsack problem and argue about its correctness.

Problem 2

Solved Exercise 3 from Kleinberg and Tardos - Algorithm Design, Chapter 4, page 187.

Suppose you are given a connected graph G, wieh edge costs that you may assume are all distinct. G has n vertices and m edges. A particular edge e of G is specified. Give an algorithm with running time $O(m+n)$ to decide whether e is contained in a minimum spanning tree of G.

Problem 3

A contiguous subsequence of length k a sequence S is a subsequence which consists of k consecutive elements of S. For instance, if S is $1, 2, 3, -11, 10, 6, -10, 11, -5$, then $3, -11, 10$ is a contiguous subsequence of S of length 3. Give an algorithm based on dynamic programming that, given a sequence S of n numbers as input, runs in linear time and outputs the contiguous subsequence of S of maximum sum. Assume that a subsequence of length 0 has sum 0. For the example above, the answer of the algorithm would be $10, 6. -10, 11$ with a sum of 17.