## Assignment 4

Due: February 18, 2020

Your solutions must be typed (preferably typeset in LATEX) and submitted as a hard-copy at the beginning of class on the day its due. When asked to provide an algorithm you need to give well formatted pseudocode, a description of how your code solves the problem, and a brief argument of its correctness.

**Problem 1: Maximum Subarray Sum** The Maximum Subarray Sum problem is the task of finding the contiguous subarray with largest sum in a given array of integers. Each number in the array could be positive, negative, or zero. For example: Given the array [-2, 1, -3, 4, -1, 2, 1, -5, 4] the solution would be [4, -1, 2, 1] with a sum of 6.

(a) [5 points] Give a brute force solution for this problem with complexity of  $O(n^2)$ .

(b) [10 points] Give a divide and conquer solution for this problem with complexity  $O(n \log n)$ .

(c) [10 points] Give a dynamic programming solution for this problem with complexity O(n).

**Problem 2: Restaurant Placement** A new restaurant chain is opening and you have been given the task of selecting the restaurant locations with the goal of maximizing their total profit.

The street network is described as an undirected graph G = (V, E), where the potential restaurant sites are the vertices of the graph. Each vertex v has a non-negative integer value  $p_v$ , which describes the potential *profit* of site v. Two restaurants cannot be built on adjacent vertices. You are supposed to design an algorithm that outputs the chosen set  $U \subseteq V$  of sites that maximizes the total profit  $\sum_{v \in V} p_v$ .

For parts (a)-(c), suppose that the street network G is *acyclic*, i.e. a tree.

(a) [5 points] Consider the following greedy restaurant placement algorithm. Choose the highest profit vertex  $v_0$  in the tree, breaking ties according to some ordering on vertex names, and put it into U. Remove  $v_0$  and all of its neighbors from G. Repeat until no vertices remain. Give a counterexample to show that this algorithm does not always give a restaurant placement with maximal profit.

(b) [10 points] Give an efficient algorithm to determine a placement with maximum profit.

(c) [5 points] Suppose that, in the absence of good data, the restaurant chain decides that all sites are equally good. The goal therefore is to simply find the placement with the maximum number of locations. Give a simple greedy algorithm for this case and argue its correctness.

(d) [5 points] Suppose that the graph is arbitrary and not necessarily *acyclic*. Give the fastest correct algorithm you can for solving the problem.