CS 670 (Spring 2025) — Assignment 2

Due: 02/26/2025

(1) [10 points]
Problem 6.5 from the textbook

(2) [10 points]
Problem 6.22 from the textbook

(3) [10 points]
Here is a weighted maximization variant of the VERTEX COVER problem. You are given an undirected graph $G = (V, E)$ with $n = |V|$ vertices in which each edge e has a weight $w_e \geq 0$; you are also given an integer $k \leq n$. Your goal is to select a set $S \subseteq V$ of k vertices so as to maximize the total weight of edges that have at least one endpoint selected, i.e., maximize $r(S) := \sum_{e=(u,v):\{u,v\} \cap S \neq \emptyset} w_e$. Notice that if $w_e = 1$ for all e, then the decision version is just VERTEX COVER (can you hit all edges with k vertices?), so this problem is NP-hard in general. For that reason, we will focus on the special case where G is a tree.

Give and analyze a polynomial-time algorithm (polynomial in n, not pseudo-polynomial) that solves thie problem optimally when G is a tree. You can get partial credit if your algorithm needs to assume that the degree of each node in the tree is bounded by a constant d and has running time exponential in d (but polynomial in n).

(4) [10 points]
You want to tile the hallway of your home with rectangular tiles of size 2×1. You can use them horizontally or vertically. Your hallway is of size $k \times n$, and your tiling must not let any of those kn squares be uncovered by a tile. To illustrate tilings, two valid tilings of a 3×4 rectangle are given in Figure 1.

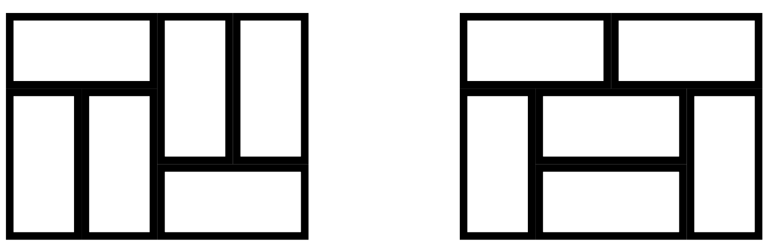

Figure 1: Two distinct tilings of a 3×4 rectangle.

Your goal is to find out how many *distinct* legal tilings there are for your hallway. If two tilings are mirrors or rotations of each other, we still count them as distinct. Give an algorithm with running time $f(k) \cdot p(n)$ for this problem, where $p(n)$ must be a polynomial, while $f(k)$ is allowed to be (singly) exponential.

(Hint: start by thinking about small values of k, like $k = 2$ or $k = 3$.)

(5) [0 points]
Chocolate Problem (1 chocolate bar): You own a long highway, with n source nodes $s_1, s_2, \ldots, s_n$ (in order), and one sink $t = s_{n+1}$. At each node s_i, there is a person i with a budget b_i. All people want to go to the sink. For each road segment from s_j to s_{j+1}, you get to set a (non-negative) price p_j. In order to go to t, a driver has to pay you the sum of all prices p_j of road segments they use, so the driver at s_i pays $\sum_{j=i}^{n} p_j$. If a driver's budget is less than the total they would have to pay, they will stay home and not drive at all, and you get no money from them.

Your goal is to find a vector of prices p_j for the segments which maximizes your total revenue. Give and analyze a polynomial-time algorithm (not pseudo-polynomial) for this problem.