HW 3

Due on **Wednesday, September 15, 2021 at 10am**

CS 473: Algorithms, Fall 2021 Version: **1.2**

Submission guidelines and policies as in homework 1.

7 (100 PTS.) Independent coloring.

(You do not need to provide pseudo-code for this question as this is pretty tedious - a detailed and concise description of your algorithm, and a careful analysis of your algorithm correctness and running time would suffice.)

7.A. (50 PTS.) You are given a graph $\mathsf{G} = (\llbracket n \rrbracket, \mathsf{E})$, where $\llbracket n \rrbracket = \{1, \ldots, n\}$. Two vertices i, j in G might be connected only if $|i - j| \leq \alpha$, where α is some small integer constant. Describe a polynomial time algorithm that computes the smallest vertex cover in G. (Your algorithm needs to output the optimal vertex cover – describe in detail how to do this.) How fast is your algorithm?

(Hint: First solve the problem when $\alpha = 1$, then for $\alpha = 2$, $\alpha = 3$, etc.)

7.B. (50 PTS.) For the graph given above, describe an algorithm, as fast possible, that computes a coloring of G with a minimum number of colors. To make things simpler, it is enough if your algorithm outputs the minimum number of colors used by a valid coloring – there is no need to output the coloring itself. How fast is your algorithm?

(As this part is similar to the previous part, you can just **sketch** your solution here – no need to repeat ideas/analysis/etc already done. Only the running time analysis needs to be done in detail.)

8 (100 PTS.) Edit distance with non-uniform cost.

Consider the edit distance problem where the price of i consecutive deletions, or i consecutive insertions has cost $f(i)$ (where f is a function provided to you in advance, which you can look up its value for specific i in constant time). To keep things simple, matching two different letters still has unit cost, and matching two identical letters cost zero.

For example, for $f(i) = 1 + i^2$, consider the following alignment and its corresponding edit distance price:

a	l	g		o	r	i	thm	
a	l	g	e	b	r	a		
0	0	0	2	1	0	1	10	total price: 12

In this model, any two or more consecutive insert (or delete) operations are merged into a longer operation (as in this example). Note, that f is arbitrary, and $f(i)$ can be smaller than 1, and f does not need to be a monotone function. (As such, the edit sequence of operations is not allowed to have two consecutive inserts, or two consecutive deletes.)

Given two input strings $S[1 \ldots n]$ and $T[1 \ldots m]$, describe an algorithm, as fast as possible, that computes the minimum edit distance in the above pricing model.

9 (100 PTS.) Walk but stay connected.

You are given n disks $D_1, \ldots, D_n$ in the plane, and a sequence $p_1, \ldots, p_m \in \mathbb{R}^2$ of points. The disk D_i represents the region that can be served by the ith server (e.g., a cellular antenna). The customer starts at p_1, and in each time unit, either it moves to p_{i+1}, or it stays put. The customer must be connected at any point in time to a server. As such, if the customer is at location p_i and is served by the disk D, then it can move next to p_{i+1} if and only if $p_i, p_{i+1} \in D$. Alternatively, if the customer is at location p_i, it can stay put, and change the current server (that it is connected to), to a different server D' – this requires that $p_i \in D'$.

Describe an algorithm, as fast as possible, that decides if there is a valid way for the customer to travel all the points in the sequence (in their order along the sequence), so that all the moves are valid. What is the running time of your algorithm?