Instructions

- please check HW1-3 for detailed instructions (they remain the same)

19 (100 PTS.) Self edit distance (take 2).

You are given a string $S = s_1 s_2 \ldots s_n$ of length n over a finite alphabet Σ. You are also given a budget $k \leq \lceil n/2 \rceil$. Describe an algorithm, as fast as possible, that computes two subsequences U, V of S, such that (i) the edit distance between U and V is at most k (under the restriction specified below), and (ii) the total length $|U| + |V|$ is maximized. Here, every edit operation has cost 1. For example, for `urbana-banananan` with a budget $k = 2$, a solution might be the subsequences

u*rbana*-*bananananan*.

As the edit distance between *rbananana* and *bananan* is two. The value of this solution is 16. (This is not the optimal solution in this case.)

Importantly, the two subsequences are not necessarily disjoint, but you are not allowed to match an original character to itself in the edit distance (thus, you can not just match the whole input string to itself).

20 (100 PTS.) Trees have needs, but then who doesn't?

We are given a tree $T = (V, E)$ with n vertices. Assume that the degree of all the vertices in T is at most 3. You are given a function $f : V \to \{0, 1, 2, 3\}$. The task is to compute a subset X of edges, such that for every node $v \in V$, there are at least $f(v)$ distinct edges in X that are adjacent to v.

Describe an algorithm, as fast as possible, that computes the minimum size set $X \subseteq E$ that meets the needs of all the nodes in the tree. The algorithm should output both $|X|$ and X itself.

21 (100 PTS.) Trust but shadow.

A classical tactic in war, is for a small force to "shadow" a bigger force of the enemy. So, assume we have a main army moving along a sequence $p_1, \ldots, p_n$ of locations. Initially, the army start at p_1, and every day it moves forward to the next location, spending the night there. The shadow army, similarly, knowing their enemy path, has a sequence of locations $q_1, q_2, \ldots, q_m$ that it is planning to travel through.

The shadow army, being much smaller, can move much faster than the main army. In particular, it can move through as many locations as it wants in one day.

The important thing with shadowing, is that the shadow army should not be too close to the main army when they both camp at night (because that would trigger a battle, which would be bad). Similarly, the shadow army should not be too far from the main army, as then it can not keep track of it.

The task at hand is to come up with a schedule for the shadow army. In the beginning of ith day, the main army is at location p_i, and the shadow army is at location $q_{\pi(i)}$ (π is what you have to compute). We require that $\pi(1) = 1$, $\pi(i+1) \geq \pi(i)$, and $\pi(n) = m$. The locations $p_1, \ldots, p_n$ and $q_1, \ldots, q_m$ are points in the plane (and are the input to the algorithm), and the distance between two locations is the Euclidean distance between them.

21.A. (40 PTS.) An interval $[x, y] \subseteq \mathbb{R}$ is ***feasible*** if there exists a valid schedule π, such that, for all i, we have $\|p_i - q_{\pi(i)}\| \in [x, y]$. Given such an interval $[x, y]$, describe a dynamic program algorithm, as fast as possible, that uses as little space as possible, that decides if $[x, y]$ is feasible (no need to output the schedule).

21.B. (30 PTS.) Describe an algorithm, as fast as possible, that computes the maximal x, such that the interval $[x, \infty]$ is feasible.

21.C. (30 PTS.) The ***instability*** of an interval $[x, y]$, with $0 < x < y$, is the ratio y/x. Describe an algorithm, as fast as possible, that computes the interval with minimum instability, among all feasible intervals.