HW 6

Due on **Wednesday, October 12, 2022 at 9am CST**

CS/ECE 374A: Intro. Algorithms & Models of Computation, Fall 2022 Version: **1.0**

Submission instructions as in previous homeworks.

Instructions for dynamic programming. In any dynamic programming solution, you should follow the steps below (in this order):

(I) Provide a clear and precise definition of the subproblems (i.e., what the recursive function is computing).

(II) Derive a recursive formula to solve the subproblems (including base cases), with justification or proof of correctness if the formula is not obvious.

(III) Specify a valid evaluation order.

(IV) Give pseudocode to evaluate your recursive formula bottom-up (with loops instead of recursion).

(V) Analyze the running time and space used.

Do not jump to pseudocode immediately. Never ever skip step (I)!

11

(100 PTS.) Fire in the prairie.

In Champaign county, the streets are straight, the corn is tall, and the fields are vast. Consider a long straight green street with n houses located along it at locations $x_1 < x_2 < \cdots < x_n$. There are two fire trucks initially located at the first house (i.e., x_1). At time t, for $t = 1, \ldots, m$, a request arrives at house number $\ell_t \in [\![n]\!] = \{1, \ldots, n\}$ for help. At this point, one of the trucks has to move to house ℓ_t (located at x_{ℓ_t}). If the truck is being moved from house numbered p, the price of moving it is $g(p, \ell_t)$, where $g(p,q) = 1 + (x_p - x_q)^2$ if $p \neq q$, and $g(p,q) = 0$ if $p = q$. You have a choice which one of the two trucks to move. Note, that trucks move only upon requests, and only one truck can be moved at each point in time – and it can move only to the requested house (i.e., a truck is always located at the last request it fulfilled).

Describe a dynamic programming algorithm, as efficient as possible, that computes the minimum price of moving the trucks and fulfilling the requests for help, assuming that the sequence $\ell_1, \ldots, \ell_m$ is provided in advance.

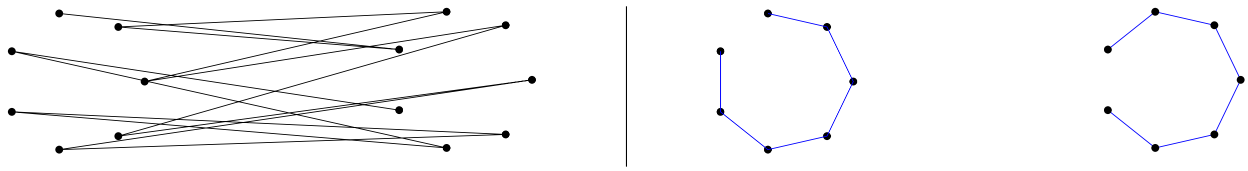
Your dynamic programming solution should use as little space as possible – specifically, how much space does your program use?

Provide a short[1] sketch of how to extend your algorithm so that it computes the optimal solution if k trucks are being used instead of two. What would be the running time of your algorithm in this case?

[1] Really short – one or two sentences.

12 (100 PTS.) Splitting into two curve.

Given a sequence of n disjoint points in the plane, $p_1, \ldots, p_n$, a ***subsequence*** I is a set of indices $1 \leq i_1 < i_2 < \cdots < i_t \leq n$. Such a subsequence defines a natural curve $\gamma(I) = p_{i_1} p_{i_2} \ldots p_{i_t}$. The length of such a curve is $\ell(I) = \|\gamma(I)\| = \sum_{u=1}^{t-1} \|p_{i_{u+1}} - p_{i_u}\|$, where $\|p - q\|$ denotes the Euclidean distance between the points p and q. Below is an example of an input curve, and the resulting split curves:



Describe a dynamic programming algorithm, as fast as possible, that computes the partition of the input sequence into two disjoint subsequences I and J, such that $\ell(I) + \ell(J)$ is minimized, and each integer $i \in [\![n]\!]$ appears exactly once – either in I or in J.

What is the running time of your algorithm? Your dynamic programming solution should use as little space as possible – specifically, how much space does your program use?

Provide a short sketch of how to extend your algorithm so that it computes the optimal such splitting into k curves. What would be the running time of your algorithm in this case?