HW 1: Grad Analysis of Algorithms (CSCI 511) Spring 2025

Goal: Understand and be able to implement dynamic programming algorithms

Remember, you may work with your classmates but you must write up your own solutions and not copy each other. Show your work! **State your group members or that you worked alone.**

Show clear, concise solutions that are a combination of pseudocode and prose. Justify the correctness and analyze the running time of your algorithms. **Read the rubric on Canvas.**

Total points: 70

1. *[30 points]* Suppose you're trying to start up a consulting business and you have clients in Seattle and Philadelphia. You're trying to minimize costs so you don't maintain offices in both places, instead you rent on a monthly basis in either Seattle or Philadelphia. Specifically, in month i, you'll incur an operating cost of S_i if you operate out of Seattle and a cost of P_i if you operate out of Philadelphia. However, if you operate out of one city in month k and the other city in month $k+1$ then you'll incur a moving cost of M to switch locations. For a sequence of n months, a plan is a sequence of n locations (each one either Seattle or Philadelphia). The cost of a plan is the sum of the operating costs for each of the n months plus the moving cost each time you switch cities. The plan can begin in either city.

Given: Two sequences of operating costs $S_1, S_2 \ldots, S_n$ and $P_1, P_2, \ldots, P_n$ and a moving cost M.

Goal: Give an efficient algorithm to calculate the cost of an optimal plan and list a plan that achieves the minimum cost.

Example: Let $M = 10$, $n = 4$, $S = [1, 3, 12, 10]$ and $P = [9, 6, 4, 6]$. In this case the optimal plan would be [Seattle, Seattle, Philly, Philly] with a cost of $1 + 3 + 4 + 6 + 10 = 24$.

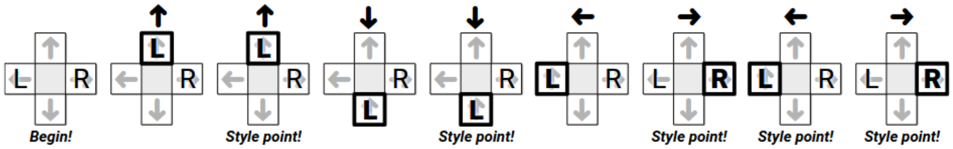
Credit for this problem to Wesley Deneke.

2. *[40 points]*

Dance Dance Revolution is a dance video game, first introduced in Japan by Konami in 1998. Players stand on a platform marked with four arrows, pointing forward, back, left, and right, arranged in a cross pattern. During play, the game plays a song and scrolls a sequence of n arrows ($\leftarrow$, $\rightarrow$, $\uparrow$, or $\downarrow$) from the bottom to the top of the screen. At the precise moment each arrow reaches the top of the screen, the player must step on the corresponding arrow on the dance platform. (The arrows are timed so that you'll step with the beat of the song.)

You are playing a variant of this game called "Vogue Vogue Revolution", where the goal is to play perfectly but move as little as possible. When an arrow reaches the top of the screen, if one of your feet is already on the correct arrow, you are awarded one style point for maintaining your current pose. If neither foot is on the right arrow, you must move one (and only one) foot from its current location to the correct arrow on the platform. If you ever step on the wrong arrow, or fail to step on the correct arrow, or move more than one foot at a time, or move either foot when you are already standing on the correct arrow, all your style points are taken away and you lose the game.

How should you move your feet to maximize your total number of style points? For purposes of this problem, assume you always start with your left foot on $\leftarrow$ and your right foot on $\rightarrow$, and that you've memorized the entire sequence of arrows. For example, if the sequence is $\uparrow\uparrow\downarrow\downarrow\leftarrow\rightarrow\leftarrow\rightarrow$ you can earn 5 style points by moving your feet as shown below:



   1. *[10 points]* Prove that for any sequence of n arrows, it is possible to earn at least $n/4 - 1$ style points.
   2. *[30 points]* Describe an efficient algorithm to find the maximum number of style points you can earn during a given VVR routine. The input to your algorithm is an array `Arrow[1:n]` containing the sequence of arrows.

Credit for this problem: Jeff Erickson