University of Toronto at Scarborough

CSCC73 - Algorithm Design and Analysis

Assignment #4: Dynamic Programming 1

Due: Feb 11, 2023 at 11.59pm This exercise is worth 5% of your final grade.

Warning: Your electronic submission on MarkUs affirms that this exercise is your own work and no one else's, and is in accordance with the University of Toronto Code of Behaviour on Academic Matters, the Code of Student Conduct, and the guidelines for avoiding plagiarism in CSCC73. Late assignments will not be accepted. If you are working with a partner your partners' name must be listed on your assignment and you must sign up as a "group" on Gradescope. Recall you must not consult **any outside sources except your partner, textbook, TAs and instructor**.

1. With all your school work you're running out of time to complete simple tasks like laundry. Since you are a poor student, you realize that if you're going to outsource your laundry there are only two ways you can afford to do this. Either you ask your mom - who immediately tells you to "grow up" or you get together with friends to organize a group laundry service. You poll all your friends to determine who wants in on which weeks and how many loads they each have. From this you have a weekly schedule for n weeks where each value is a number of loads ℓ_i for week i. You research the two closest laundromats that offer pick-up and delivery. Laundromat L_1 and laundromat L_2 have different methods for charging for their services.

   - L_1 charges a fixed rate r per load (so it costs $r \cdot \ell_i$ to pick-up, wash and deliver a week's load ℓ_i.)
   - L_2 makes contracts for a fixed amount w per week, independent of the number of loads. However, contracts with company L_2, must be made in blocks of three consecutive weeks at a time.

   A *schedule*, for you and your friends laundry, is a choice of laundromat (L_1 or L_2) for each of the n weeks, with the restriction that laundromat L_2, whenever it is chosen, must be chosen for blocks of three contiguous weeks at a time. The *cost* of the schedule is the total amount paid to laundromat L_1 and L_2, according to the description above.

   Give a polynomial-time algorithm that (makes you the laundry hero and) takes a sequence of weekly load values $\ell_1, \ell_2, \ldots \ell_n$ and returns a *schedule* of minimum cost. Explain your algorithm and justify your recurrence relation and complexity. A formal proof is not necessary.

   Example Suppose $r = \$10$, $w = \$80$, and the sequence of values is

   $$5, 8, 10, 11, 9, 6, 7.$$

   Then the optimal schedule would be to choose laundromat L_1 for the first two weeks, then laundromat L_2 for a block of three consecutive weeks and then laundromat L_1 for the final three weeks.

2. Suppose that you are given a sequence S of price quotes for a particular stock. You wish to locate trends in the stock price over time. In particular, you wish to find the longest trend in S that is strictly increasing. Design an algorithm to find the *longest increasing trend* of S. For example, if the sequence S is

   $$S = 0.45, 0.43, 0.42, 0.46, 0.47, 0.52, 0.49, 0.50, 0.51, 0.53, 0.48, 0.44$$

   a longest increasing trend is 0.42, 0.46, 0.47, 0.49, 0.50, 0.51, 0.53 where as 0.45, 0.46, 0.47, 0.52, 0.53 is not.