Homework 2.
Due: Monday, September 12, 2022 before 8am EDT.

[DPV] Practice Dynamic Programming Problems

[DPV] Problem 6.17 – Making-change I

Given an unlimited supply of coins of denominations $x_1, x_2, ..., x_n$, we which to make change for a value v...

[DPV] Problem 6.18 – Making change II

Consider the following variation on the change-making problem (Exercise 6.17): you are given denominations $x_1, x_2, ..., x_n$, ...

[DPV] Problem 4.11 (length of shortest cycle)

Give an algorithm that takes as input a directed graph with positive edge lengths, and returns the length of the shortest cycle in the graph (if the graph is acyclic, it should say so). Your algorithm should take time at most $O(|V|^3)$.

[DPV] 4.21 (Currency trading)

Shortest path algorithms can be applied in currency trading. Let $c_1, c_2, \ldots, c_n$ be various currencies. . .

[DPV] Practice Divide and Conquer Problems

[DPV] Problem 2.1– Practice multiplication.
[DPV] Problem 2.5 – Solving recurrence.
[DPV] Problem 2.7 – Sum and product of roots of unity.
[DPV] Problem 2.14 – removing duplicates.

See next page for homework problems.

DP Homework (graded)

Problem 1 (Road Trip)

You'd like to take a road trip from checkpoint c_0 to checkpoint c_n, but your car can only go 300 miles at a time before needing to refuel. Fortunately, there are checkpoints at specific positions on your path where you can refuel, $C = [c_1, c_2, \ldots, c_n]$, which you can think of as points on a number line. However, each checkpoint has an associated "penalty" p_i that it will cost you to use its gas pump (curiously, this is a one-time fee per checkpoint, and doesn't depend on how much gas you get). c_0 is always at location 0 along the path, c_n is always reachable by some series of checkpoints, and we always refuel at c_n. We want to know the minimum total penalty to get to (and refuel at) checkpoint c_n.

For example, if we have $C = [100, 250, 400]$, and $P = [12, 20, 3]$, then we can go from c_0 to checkpoint c_1 and then to checkpoint c_3 for a total penalty of $12 + 3 = 15$, which is the minimum penalty to get to c_3. We could travel further initially by going from c_0 straight to c_2 (and then to c_3), but this isn't optimal as it costs us total penalty $20 + 3 = 23$.

Design a dynamic programming algorithm to, given C and P, efficiently find the minimum total penalty it will cost you to get from c_0 to c_n.

(a) Define the entries of your table in words. E.g., $T(i)$ or $T(i, j)$ is ....

(b) State recurrence for entries of table in terms of smaller subproblems.

(c) Write pseudocode for your algorithm to solve this problem.

(d) Analyze the running time of your algorithm.

Problem 2 (Finding an index)

Given an array $A[1 \dots n]$ whose elements are odd numbers (positive and negative). Suppose that n is a power of 2, the elements in $A[1 \dots n]$ are sorted and distinct. Design an algorithm to check whether there is at least one element in the array satisfies $A[i] = 2i + 5$, where $1 \leq i \leq n$ is the index of the i^{th} element in the array. If such element exists, your algorithm should return "yes"; otherwise, return "no". The running time of your algorithm is required to be $O(\log(n))$. Describe your algorithm with words (no pseudocode) and explain why it is correct. Justify the running time of your algorithm.