CS 336 Assignment 4

February 21st 2025

Problem 1.

You want to travel from city A to city B located on a straight line (A is located in position 0 and B is located in position $M \geq 0$), and you can travel at most distance $D \geq 0$ miles per day, and you can only move to the right. You have hotels between A and B with locations $a_1, \ldots, a_n$, where you can stay for a night.

You are a person who likes to optimize all aspects of your life. In particular, if you didn't fully use all D miles per day, it causes you great distress. Namely, if on some day you traveled distance d miles (out of possible D miles), the amount of distress is 2^{D-d}.

You start at city A. At the end of each day you have to be at some hotel or city B. Your goal is to reach city B while suffering the least total amount of distress.

Example: Assume that $D = 4$ and city B is located in position 6. You have two hotels in locations 2 and 3. The following routes have the following distress:

- $0 \to 2 \to 6$: $2^{4-(2-0)} + 2^{4-(6-2)} = 4 + 1 = 5$
- $0 \to 2 \to 3 \to 6$: $2^{4-(2-0)} + 2^{4-(3-2)} + 2^{4-(6-3)} = 4 + 8 + 2 = 14$
- $0 \to 3 \to 6$: $2^{4-(3-0)} + 2^{4-(6-3)} = 2 + 2 = 4$

The last route is optimal.

Please do the following:

- Formulate the subproblem. Please state it as precisely as possible.
- Design a dynamic programming algorithm for solving this problem:
  - State the base case.
  - State the recurrence relation.
  - Explain why the recurrence relation is correct (from your explanation, one should understand how to get your the recurrence relation).
  - Please provide the **pseudocode**. Please use the bottom-up approach.
  - Explain:
    - What is the running time of your algorithm (all arithmetic operations take constant time). **The running time must be as small as possible.**
    - How to recover the maximum reward.

Problem 2.

Alice has a perfect forecast of stock prices for the next n days. For each day i, Alice can do **at most one** action from the following list:

- Buy one stock of price b_i.
- Sell one stock at price s_i. She may choose this option only if she has at least one stock.

Due to anti-monopoly laws, Alice cannot possess more than k stocks at any time. What is the maximum profit Alice can make by the end of day n?

Example: Assume that $n = 4$ and $k = 1$ and we have $(b_1 = 1, s_1 = 4)$, $(b_2 = 3, s_2 = 2)$, $(b_3 = 3, s_3 = 100), (b_4 = 1, s_4 = 10)$. The optimal solution is to buy on day 1 and sell on day 3, giving profit $100 - 1 = 99$. Note that the following solutions are not allowed:

- Buy on days 1 and 2, sell on days 3 and 4. After day 2, she holds 2 stocks, which is greater than k.
- Buy on day 1 and sell on day 3, then buy on day 4 and sell on day 4. She performs two actions on day 4.

Please implement the following function.

```
vector<int> Stocks(const vector<int>& buy, const vector<int>& sell, int k)
```

The function should output the vector of length n, where for each day, the corresponding value of the vector should be:

- 1 if you buy a stock on that day;
- -1 if you sell a stock on that day;
- 0 if you don't do anything on that day.

It is guaranteed that the solution is unique for all tests.

Time limit The instructions are similar to the previous programming assignments. Your program should pass each tests in no more than 1 second. You can assume that $0 \le n \le 7000$, $0 \le k \le 10^9$, and all other numbers are between 0 and 10^9.

Memory limit You have 1.5 GB of memory available. If your solution works on your machine but fails when you submit it to Gradescope, check your memory.