Wall Street Bets

This is a **regular task**. You must submit a PDF, which can be produced using the LaTeX template on Moodle, exported from a word processor, hand-written or any other method.

Satoshi is an investor trying to maximise their earnings from their new investment *Algocoin*. For each of the next n days, Satoshi knows that if they invest on day i, they will get exactly $C_i \geq 0$ dollars. However, investing on *consecutive* days (i and $i+1$) too many times will lock Satoshi out of their *Algocoin* account.

(a) Currently, *Algocoin* allows for investing on consecutive days *exactly once* over the n days. Design an efficient dynamic programming algorithm to determine the maximum investment profit for Satoshi, and justify its correctness and time complexity.

For example, suppose $n = 6$ and $C = [15, 14, 13, 16, 12, 13]$.

- The optimal strategy is to invest on days 1, 2, 4 and 6, for a total of 58 dollars.
- Investment on days 1, 2, 3 and 5 is not allowed, as this sequence includes two consecutive investments.
- Likewise, investment on days 1, 2, 3 and 5 is not allowed, as this sequence includes two consecutive investments.
- Investment on days 1, 2, 4 and 5 is similarly not allowed, as this sequence also includes two consecutive investments.

(b) *Algocoin* has been improved and now permits consecutive investments up to k times where $1 < k < n$. Using part (a) or otherwise, design and analyse an efficient algorithm to solve this variant of the problem.

Advice.

Solutions for both parts should include a well-defined subproblem definition, base case(s), recurrence, order of computation, final solution and correctness (as with the advice for previous weeks' dynamic programming solutions).

You may elect to just do part (b), since (a) is just a special instance of (b) where $k = 1$. However, as with other tasks, it is instructive to do part (a) first especially if you are struggling with how to get started.

Expected length: Up to two pages in total.