

## Homework #8

You should try to solve these problems by yourself. I recommend that you start early and get help in office hours if needed. If you find it helpful to discuss problems with other students, go for it. **You do not need to turn in these problems. The goal is to be ready for the in class quiz that will cover the same or similar problems.**

### Problem 1: Dynamic Programming

On a staircase, the  $i$ -th step has some non-negative cost  $\text{cost}[i]$  assigned (0 indexed). Once you pay the cost, you can either climb one or two steps. You need to find minimum cost to reach the top of the floor, and you can either start from the step with index 0, or the step with index 1.

Example: Input:  $\text{cost} = [10, 15, 20]$  Output: 15

A function  $C(i)$  is defined to denote the minimum cost from the first step to the  $i$ th step. Provide  $C(1)$  (i.e., only the first step is considered) and  $C(2)$  (i.e., only the two steps are considered). Then provide  $C(i)$  in terms of previously computed values of  $C$ .

### Problem 2: Knights

Given an  $n \times n$  chessboard and a knight that starts at position  $A1$ , devise an algorithm that returns how many ways the knight can end up at position  $xy$  after  $k$  moves. Knights move  $\pm 1$  squares in one direction and  $\pm 2$  squares in the other direction. (Hint: let  $N(k, x, y)$  denote the number of ways for a knight to end up on space  $x, y$  after  $k$  moves).

### Problem 3: Moving in a Grid

Imagine that you are placed on a grid with  $n$  spaces in every row and  $n$  spaces in every column. You can start anywhere along the bottom row of the grid, and you must move to the top row of the grid. Each time you move, you can either move directly up (staying in the same column, but moving up a row), up and to the left (moving over one column and up one row), or up and to the right (moving over one column and up one row). You cannot move up and to the left if you are in the leftmost row, and you cannot move up and to the right if you are in the rightmost row.

Each time you move, you are either paid or pay; that is, every legal move from square  $x$  to square  $y$  is assigned a real value  $p(x, y)$ . Sure,  $p(x, y)$  can also be 0.

Give a dynamic programming algorithm to compute your sequence of moves to receive the maximum payoff to move from the bottom of the grid to the top of the grid. (Your maximum payoff may be negative.) You must calculate the *value* of the optimal solution (i.e., the payoff) *and* the solution itself (i.e., the sequence of moves). Again, you can start at any square in the bottom row and end in any square in the top row.