

### 1. Moving on a Checkerboard

Suppose that you are given an  $n \times n$  checkerboard and a checker. You must move the checker from the bottom edge of the board to the top edge according to the following rule. At each step you may move the checker to one of three squares:

- 1) the square immediately above
- 2) the square that is one up and one to the left (but only if the checker is not already in the leftmost column)
- 3) the square that is one up and one to the right (but only if the checker is not already in the rightmost column)

Each time you move from square  $x$  to square  $y$ , you receive  $p(x, y)$  dollars. You are given a list of the values  $p(x, y)$  for each pair  $(x, y)$  for which a move from  $x$  to  $y$  is legal. Do not assume that  $p(x, y)$  is positive.

Give an algorithm that figures out the set of moves that will move the checker from somewhere along the bottom edge to somewhere along the top edge while gathering as many dollars as possible. Your algorithm is free to pick any square along the bottom edge as a starting point and any square along the top edge as a destination in order to maximize the number of dollars gathered along the way. What is the running time of your algorithm?

### 2. Maximizing Profit

You are given lists of values  $h_1, h_2, \dots, h_k$  and  $l_1, l_2, \dots, l_k$ . For each  $i$  you can choose  $j_i = h_i$ ,  $j_i = l_i$ , or  $j_i = 0$ ; the only catch is that if  $j_i = h_i$  then  $j_{i-1}$  must be 0 (except for  $i = 1$ ). Your goal is to maximize  $\sum_{i=1}^k j_i$ .

Give an efficient algorithm that returns the maximum possible value of  $\sum_{i=1}^k j_i$ .

### 3. Maximum alternating subsequence

An *alternating sequence* is a sequence  $a_1, a_2, \dots$  such that no three consecutive terms of the sequence satisfy  $a_i > a_{i+1} > a_{i+2}$  or  $a_i < a_{i+1} < a_{i+2}$ .

Given a sequence, efficiently find the longest alternating subsequence it contains. What is the running time of your algorithm?