TA Lab Week 2 – Fundamentals

The goal of this lab is to help develop your algorithmic thinking further, as you attempt to efficiently solve a number of puzzle problems.

Exercise 1 – Unique paths

Image you have a grid of `r` rows and `c` columns. You are standing at the top left corner of a grid (`s`) and you want to move to a goal (`g`). You can move through the grid only by moving right or down. For any two given integers (`r,c`), how many unique paths exist to get from `s` to `g`?

For a 2x2 grid the answer is 2. You can go down and right, or right and down:

s	
	g

What for a grid `r` x `c`**?**

Hint

What happens if you reach the border of the grid?

Exercise 2 – Min-cost climbing stairs

On a staircase, the i-th step has some non-negative cost `cost[i]` assigned to it. The staircase starts at index i=0. Once you pay the cost, you can either climb one or two steps. Design and implement an algorithm to find the minimum cost to reach the top floor. You can start your climb from either step index 0, or step index 1.

For example, given a staircase of 9 floors (from 0 to 8) and input cost = [1, 100, 1, 1, 1, 100, 1, 1, 100, 1], the min-cost climb starting from 0 is 6.

Hint

Think from top to bottom, and bottom to top.

Exercise 3 – Search in a bitonic array

An array is bitonic if it is comprised of an increasing sequence of integers followed immediately by a decreasing sequence of integers. Write a program that, given a bitonic array of *n* distinct integer values, determines whether a given integer is in the array. Use O(lg *n*) compares in the worst case.

For example, the array [2,4,6,8,10,12,11,9,7,5,3] is bitonic, while the following are not: [1,2,3], [1,2,3,2,3],[5].

Hint

First, find the maximum integer using lg *n* compares—this divides the array into the increasing and decreasing pieces.