# HW 7

Due on **Monday, March 23, 2020 at 11pm**

**CS/ECE 374 B: Algorithms & Models of Computation, Spring 2020** Version: **1.0**

**Submission instructions as in previous homeworks.**

Unless stated otherwise, for all questions in this homework involving dynamic programming, you need to provide a solution with explicit memoization.

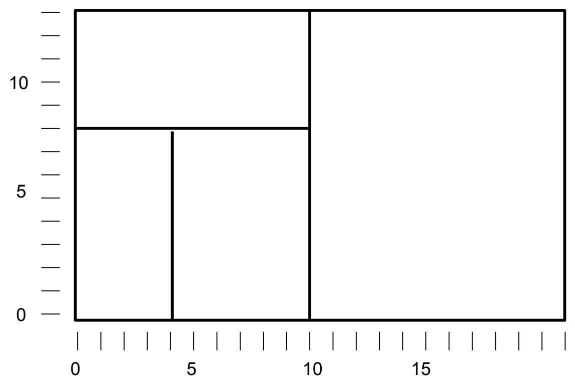
## 19 (100 PTS.) Subdividing a rectangle slab.

You have mined a large slab of marble from a quarry. For simplicity, suppose the marble slab is a rectangle measuring $n$ inches in height and $m$ inches in width. You want to cut the slab into smaller rectangles of various sizes - some for kitchen counter tops, some for large sculpture projects, others for memorial headstones. You have a marble saw that can make either horizontal or vertical cuts across any rectangular slab. At any time, you can query the spot price $P[x, y]$ of an $x$-inch by $y$-inch marble rectangle, for any positive integers $x$ and $y$. These prices depend on customer demand, and people who buy marble counter tops are weird, so don't make any assumptions about them; in particular, larger rectangles may have significantly smaller spot prices. Given the array of spot prices and the integers $m$ and $n$ as input, design a dynamic programming algorithm to compute how to subdivide an $n \times m$ marble slab to maximize your profit.

Your solution must output *both* the maximum profit *as well* as the sequence of cuts necessary to obtain that profit. A sequence of cuts can be described as a sequence of tuples $(V, x, y_{min}, y_{max})$ for vertical cuts, $(H, y, x_{min}, x_{max})$ of horizontal cuts.

For example, the rectangle depicted below has been subdivided via the sequence of cuts

$$[(V, 10, 0, 13), (H, 8, 0, 10), (V, 4, 0, 8)]$$

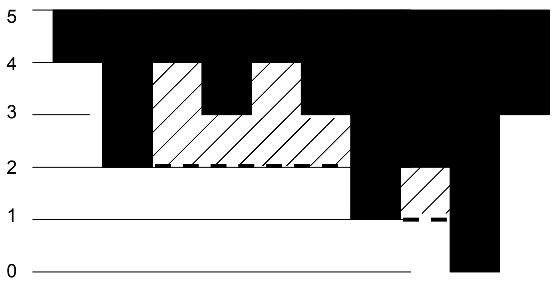


Make your solution as asymptotically efficient as you can; we are not providing any particular target.

## 20 (100 pts.) How much air?

An underground cavern near the town has flooded during a storm, trapping several unprepared hikers who were exploring at the time. An engineer at Daring Tech Solutions (DTS) comes up with a bold, though ultimately misguided plan: Without going into all the details, it involves breathing the air bubbles trapped in the rock formations making up the cavern's ceiling. To help work out if the plan is viable, DTS sends a robot to take measurements. Your goal is to estimate how much air could be trapped underneath the rocks.

The following figure illustrates the rock formation (shaded region) and the volume of air it can trap in a bubble (hatched region).



The robot is outfitted with a laser range finder, such that it can measure the height of the rocky ceiling (assume the laser passes through air and water exactly the same way, so the robot cannot measure the bubble directly).

Give a dynamic programming algorithm to compute the amount of air that can be trapped under the rocks.

To be clear about the model: Rocks and air are measured in integer units, corresponding to squares in the illustration. Range measurements are non-negative. A bubble square exists at height $h$ if and only if there exists some rock at equal height (or lower) both to the left and to the right. You can assume the left and right boundaries are of infinite height (they're holes that go up to the surface, so they don't trap air).

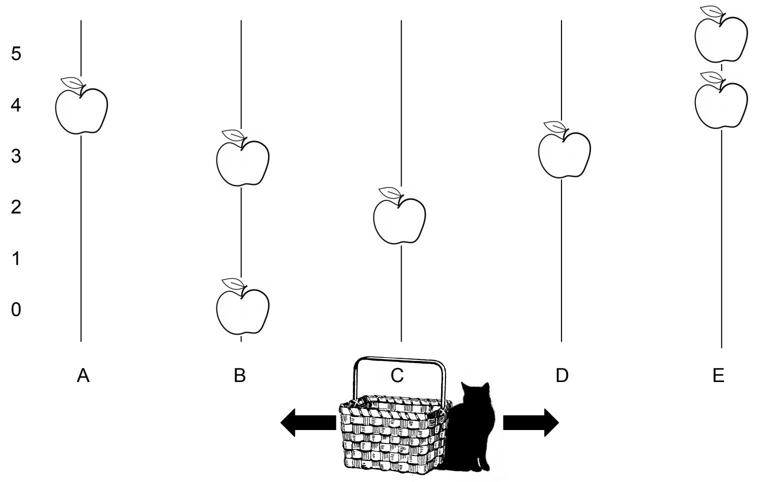
As an example, the figure above corresponds to the input

$$H = [4, 2, 4, 3, 4, 3, 1, 2, 0, 3]$$

for which your algorithm should output 7. You should aim for an algorithm that takes $O(n)$ time and $O(n)$ space, where $n$ is the length of the input array.

**21** (100 PTS.) **Basket Game.**

The hottest video game this month is BasketCat. It's a game where a cat moves a basket to catch apples that fall from the sky. Honestly I don't know how a game like this can still become popular. Anyway, it looks like this:



A "stage" of BasketCat is encoded as five sequences $A$, $B$, $C$, $D$, $E$, each containing an ordered sequence describing when apples in that position reach the ground. The basket starts out at position C.

The cat has three moves: "Left", "Right", and "Stay" (as if "stay" is a command that works on cats...). In each game step:

1. First the cat's move is applied to the basket. (If the basket is at position A, then "Left" isn't a valid move, similar with E).
2. After making a move, if the basket is in the position of an apple, then the cat earns 1 point.
3. Finally, all of the apples move down one cell.

Use dynamic programming to design an algorithm for computing the optimal strategy for a given level of BasketCat. Your solution should run in $O(n^2)$ time (no restriction on space), where $n$ is the length of the stage, the maximum value in any of $A, B, C, D, E$. The basket starts at position C.

For example, the level from the illustration corresponds to input:

$$A = [4], \quad B = [0, 3], \quad C = [2], \quad D = [3], \quad E = [4, 5]$$

for which your algorithm should output 5.