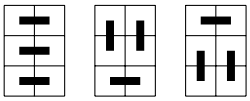
Problem Set 2

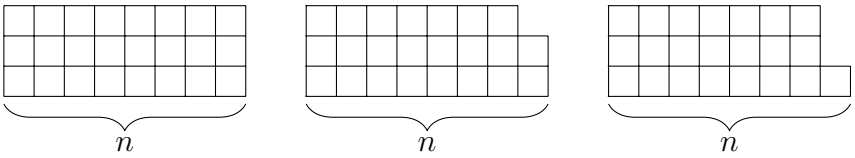
CS 331H

Due Tuesday, February 23

1. How many ways can one tile a $3 \times n$ rectangle using 2×1 tiles? For example, there are 3 ways to tile a 3×2 rectangle:



   (a) [70%] Show how to compute the answer in $O(n)$ time, assuming that the word size can represent numbers as large as the answer. You may find it useful to consider the number of ways to tile all of the following figures:



   (b) [15%] Repeat part (a) for tiling a $k \times n$ rectangle using 2×1 rectangles, for any constant k. How does your complexity scale with k?

   (c) [15%] By expressing the recursion in terms of matrices, show how to compute the part (a) answer in $O(\log n)$ time – or even $O(1)$ time, again assuming the word size can represent numbers as large as the answer.

2. You are given a sequence of n integers, $x_1, \ldots, x_n$, and an integer $k \in [n]$. Each integer is polynomially large. Find the contiguous subset of size at least k with maximum *average*. That is, find two indices $s, t \in [n]$ with $t \geq s + k - 1$ that maximize

$$\frac{1}{t - s + 1} \sum_{i=s}^{t} x_i.$$

   Full credit requires $O(n \log n)$ time. **Hints:** First, can you compute the maximum *sum* in $O(n)$ time? Second, can you determine *whether the maximum average is nonnegative* in $O(n)$ time?

3. A subsequence of an array is said to be *bitonic* if it increases up to some point and then decreases, or vice versa. Design a dynamic programming algorithm to solve the longest bitonic subsequence problem in $O(n^2)$ time (for 80% credit) or $O(n \log n)$ time (for 100% credit). You may assume that the input contains n numbers, all distinct.

4. You are given a three dimensional object. On the horizontal plane it is an $n \times n$ square, and on the vertical axis each square (x, y) is a square pillar rising to height $h_{x,y} \geq 1$. Adjacent pillars, even ones sharing corners, are fused together.

   You submerse this object into a bucket of water, then carefully lift it out. Water will then drain off the sides, but it cannot drain through pillars. How many units of water will be captured in the object? Give an $O(n^2 \log n)$ algorithm.

   As an example, in the following grid 2 units will be captured, all in the center tile:

0	5	9
7	3	6
7	5	2