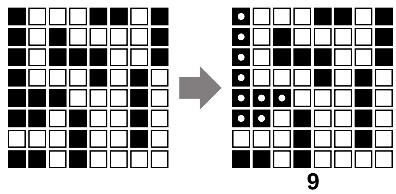
CS 473: Undergraduate Algorithms, Spring 2010
Homework 6

Written solutions due Tuesday, March 16, 2010 at noon

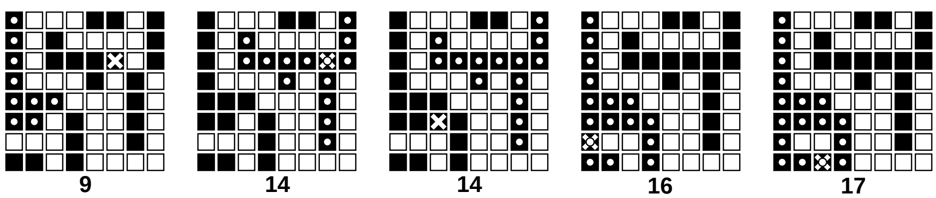
---

1. (a) Describe and analyze an algorithm to compute the size of the largest connected component of black pixels in an $n \times n$ bitmap $B[1..n, 1..n]$. For example, given the bitmap below as input, your algorithm should return the number 9, because the largest conected black component (marked with white dots on the right) contains nine pixels.



(b) Design and analyze an algorithm Blacken(i, j) that colors the pixel $B[i, j]$ black and returns the size of the largest black component in the bitmap. For full credit, the *amortized* running time of your algorithm (starting with an all-white bitmap) must be as small as possible.

For example, at each step in the sequence below, we blacken the pixel marked with an X. The largest black component is marked with white dots; the number underneath shows the correct output of the Blacken algorithm.



(c) What is the *worst-case* running time of your Blacken algorithm?

2. Suppose you are given a graph G with weighted edges and a minimum spanning tree T of G.

(a) Describe an algorithm to update the minimum spanning tree when the weight of a single edge e is *increased*.

(b) Describe an algorithm to update the minimum spanning tree when the weight of a single edge e is *decreased*.

In both cases, the input to your algorithm is the edge e and its new weight; your algorithms should modify T so that it is still a minimum spanning tree. *[Hint: Consider the cases $e \in T$ and $e \notin T$ separately.]*

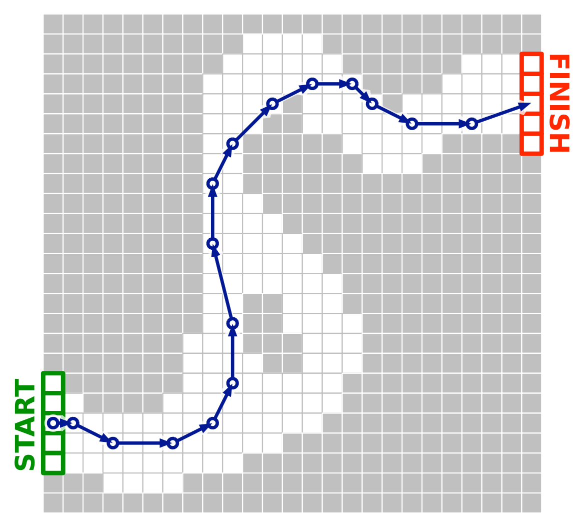
3. ***Racetrack*** (also known as *Graph Racers* and *Vector Rally*) is a two-player paper-and-pencil racing game of uncertain origin that Jeff played on the bus in 5th grade.[1] The game is played using a racetrack drawn on a sheet of graph paper. The players alternately choose a sequence of grid points that represent the motion of a car around the track, subject to certain constraints explained below.

   Each car has a *position* and a *velocity*, both with integer x- and y-coordinates. The initial position is an arbitrary point on the starting line, chosen by the player; the initial velocity is always $(0,0)$. At each step, the player optionally increments or decrements either or both coordinates of the car's velocity; in other words, each component of the velocity can change by at most 1 in a single step. The car's new position is then determined by adding the new velocity to the car's previous position. The new position must lie inside the track; otherwise, the car crashes and that player immediately loses the race. The first car that reaches a position ***on*** the finish line is the winner.

   Suppose the racetrack is represented by an $n \times n$ array of bits, where each 0 bit represents a grid point inside the track, each 1 bit represents a grid point outside the track, the 'starting line' is the first column, and the 'finish line' is the last column.

   Describe and analyze an algorithm to find *the minimum number of steps* required to move a car from the starting line to the finish line according to these rules, given a racetrack bitmap as input. *[Hint: Build a graph. What are the vertices? What are the edges? What problem is this?]*

velocity	position
$(0,0)$	$(1,5)$
$(1,0)$	$(2,5)$
$(2,-1)$	$(4,4)$
$(3,0)$	$(7,4)$
$(2,1)$	$(9,5)$
$(1,2)$	$(10,7)$
$(0,3)$	$(10,10)$
$(-1,4)$	$(9,14)$
$(0,3)$	$(9,17)$
$(1,2)$	$(10,19)$
$(2,2)$	$(12,21)$
$(2,1)$	$(14,22)$
$(2,0)$	$(16,22)$
$(1,-1)$	$(17,21)$
$(2,-1)$	$(19,20)$
$(3,0)$	$(22,20)$
$(3,1)$	$(25,21)$



A 16-step Racetrack run, on a 25×25 track. This is *not* the shortest run on this track.

[1]The actual game Jeff played was a bit more complicated than the version described in this problem. In particular, the track was a freeform curve, and by default, the entire line segment traversed by a car in a single step had to lie entirely inside the track. If a car did run off the track, it started its next turn with velocity zero, at the legal grid point closest to where it first crossed the track boundary.