**CS 374** **Spring 2015**

# Homework 4

Nikhil Unni (nunni2)

3.

```
Racetrack(grid, Starting Area, Ending Area):
  minLength = ∞
  for each (x,y) in Starting Area
      Mark each vertex u as unvisited
      length = Racetrack(x,y,0,0,0)
      if length < minLength
          minLength = length


Racetrack(x,y,dx,dy, moves):
  Q = {(x,y,dx,dy,moves)}
  while Q is not empty
      (x,y,dx,dy,moves) = next item in Q
      if (x,y) in Ending Area
          output moves

      mark (x,y) as visited
      moves = {−1, 0, 1} × {−1, 0, 1}
      for each (i,j) in moves
          if 0 ≤ (x + dx + i), (y + dy + j) ≤ 24 and grid[x+dx+i,y+dy+j] = 0 and (x+dx+i,y+dy+j) is not visited
              add (x+dx+i,y+dy+j,dx+i,dy+j, moves+1) to Q
  output ∞
```

The algorithm is just a BFS, in which I check boundary conditions for the edges of the track, or out of bounds, or if I've already visited the edge (since it's BFS, visiting a vertex on a later layer necessarily means that the path to the end will be greater than or equal to whatever previously visited the vertex). I do this with each starting point, and then pick the smallest.