CS577 Homework 3

Haoyi Lu, Fengsheng Shu

Problem 1:

Let M[1…n,…….,1…n] be the n × n array for bitmap, 1 represents that point is an obstacle and 0 represents road. We try to construct a directed graph G with O(n4) vertices and O(n4) edges (because the problem 1 says the algorithm should run in O(n4) ).

For each vertex, they have same structure as (x, y, Δx, Δy), x and y represents the current position and Δx and Δy represents the velocity. To make all vertex are legal position and velocity, we add some restriction: 1 <= x <= n, 1 <= y <= n, M[x,y] = ‘0’. For velocity, we need to calculate its largest velocity, which is and the result is v(v+1)/2, which is the total distance the car moves with the largest acceleration, so we have v(v+1)/2 <= n and get v cannot exceed . So the restriction for Δx and Δy is <= Δx <= , <= Δy <= .

We define its neighbors as (x’, y’, Δx’, Δy’) if and only if they are all legal and also x’ = x + Δx, y’ = y + Δy.

Δx’ ∈ { Δx -1， Δx ， Δx +1}, Δy’ ∈{Δy -1, Δy , Δy +1}.

G has start vertex s of the form (1,y,0,0) and finish vertex f of the form (n, y, Δx, Δy). Our goal is find the shortest path from any vertex s to any vertex f.

Now we use BFS to calculate the paths, we know the running time of BFS equals to number of vertices encountered plus number of edges traversed.

**Running time**:

For number of vertices, we have O(n2). For possible number of edges for one vertex, we have (2)2 which is 8n = O(n). So in BFS, the algorithm is :

BFS(v)

------ initialize an empty queue Q

------ add v to Q and mark v

------ set level(v) = 0

------ while (Q is non-empty) loop n2 times

W = pop(Q);

For all unmarked neighbors w’ of w loop 8n times

Add w’ to Q

Mark w’

Set level(w’) = level (w) + 1

So in total, the time complexity is n^2 \* 8n = O(n3) which is less than O(n4).