Algorithms – CS-27230

The car race problem[1]

Instructor: László Babai
e-mail: `laci@cs.uchicago.edu`

The solution should be short, **elegant**, *and convincing.*

In this problem, both space and time are **discrete:** points in space are represented by vectors with integer coordinates, and points in time are integers.

Let R be a subset of the $(n+1)^2$ points in the plane with integer coordinates between 0 and n. We call R the "race track." One of the points of R is designated as the start (S), another as the goal (G).

The points are represented as vectors (i, j).

A "car" is a particle sitting on a point at any time, say, $p_t = (i_t, j_t)$ at time t. We require that for all t we have $p_t \in R$ (the car cannot leave the race track).

We say that *time unit* t begins at time t and ends at time $t + 1$. The *velocity* of the car during this time unit is defined as the vector $v_t = p_{t+1} - p_t$.

The *acceleration/deceleration* of the car is limited by the following constraint: from any one time unit to the next one, each coordinate of the velocity can change by at most one.

For instance, if during time unit 6 the car was moving from point $p_6 = (10, 13)$ to point $p_7 = (16, 12)$ then its velocity was $v_6 = p_7 - p_6 = (6, -1)$ during this move; during the next time unit, the following are its possible velocities and corresponding destinations:

v_7 : velocity during time unit 7	p_8 : location at time 8
$(7, 0)$	$(23, 12)$
$(7, -1)$	$(23, 11)$
$(7, -2)$	$(23, 10)$
$(6, 0)$	$(22, 12)$
$(6, -1)$	$(22, 11)$
$(6, -2)$	$(22, 10)$
$(5, 0)$	$(21, 12)$
$(5, -1)$	$(21, 11)$
$(5, -2)$	$(21, 10)$

[1] Written in 2005. Last updated 2-12-2025. The problem is due to Péter Gács and László Lovász.

Of course only those locations are legal that belong to R.

During time unit 0, the car rests at Start: $p_0 = p_1 = S$. So its velocity during this time unit is $v_0 = (0, 0)$.

The objective is to decide whether the Goal is reachable at all from the Start and if so, to reach it using the minimum number of time units.

The input is the number n in unary (the string $111\dots1$ (n 1s)), the set R given as a linked list, and the points S and G. We work in the unit-cost model: $O(\log n)$-bit numbers cost one unit of space to store and one unit of time to add/subtract and to follow as an address into an array.

(a) Construct an example where the optimal route visits the same point (location) 100 times (at different velocities).

(b) Find an optimal route in $O(|R| \cdot n^2)$ time. Describe your solution in clear English statements. Pseudocode not required. Algorithms discussed and analyzed in class can be used as subroutines. Prove that your algorithm runs within the time claimed. *Hint.* Use BFS. The difficulty is in constructing the right graph to which to apply BFS. Do not overlook the possibility stated in (a).

(c) Solve the problem in $O(|R| \cdot n)$ time and space, assuming $|R| = \Omega(n)$. (The space constraint in particular requires that you do not use an array with more than $|R| \cdot n$ cells.) Point out where you are using the condition that $|R| = \Omega(n)$.

(d) Solve the problem in $O(|R| \cdot n)$ time and space. (In this problem, there is no constraint on how small R can be.)