Online Motion Planning
Problem Set 3
Universität Bonn, Institut für Informatik I

To be solved until the 15th of November

Problem 1:

a) Show that the Pledge Algorithm in a maze with n obstacle edges can visit an edge $\Omega(n)$ times, even if it finally finds an exit.

b) Suppose the robot knows from the beginning that the maze consists of n edges. Is there a number k such that the robot knows there is no exit if he already visited k edges?

Problem 2:

For the Chinese Wall problem we now use the following strategy:

> In the i. iteration:
> If i even: go right for $f(i) = 2^i$ meters and then $f(i) = 2^i$ meters to the left.
> If i odd: go left for $f(i) = 2^i$ meters and then $f(i) = 2^i$ meters to the right.

Now assume that the robot cannot execute the instructions precisely. For every Step that should have length ℓ he goes $[\ell \cdot (1 - \delta), \ell \cdot (1 + \delta)]$ for some $\delta \in [0, 1[$.

So the robot does not necessarily reach the starting point after every iteration. In the worst case, all steps to the left are too long and all steps to the right are too short resulting in a drift to the left of the supposed starting point s_i (the point to which the robot returns after the ith iteration, therefore the starting point of the $i+1$th iteration.

a) Let Δ_i denote the distance of s_i to s If $\Delta_i > 0$ then s_i lies to the left of s, if $\Delta_i < 0$ s_i lies to the right of s.
Assuming that the robot produces the same error δ' in every step, provide a formula for Δ_i.

b) What is the largest value that δ' may attain, such that the algorithm is still guaranteed to reach the goal?

Problem 3:

Consider a BUG variant in which the robot leaves an obstacle (if possible) in the direction to the goal whenever the distance to goal is growing. If it is not possible because the obstacle lies in the direction to the goal, the robot further on follows the obstacle boundary.

Does this variant reach the goal in every environment?