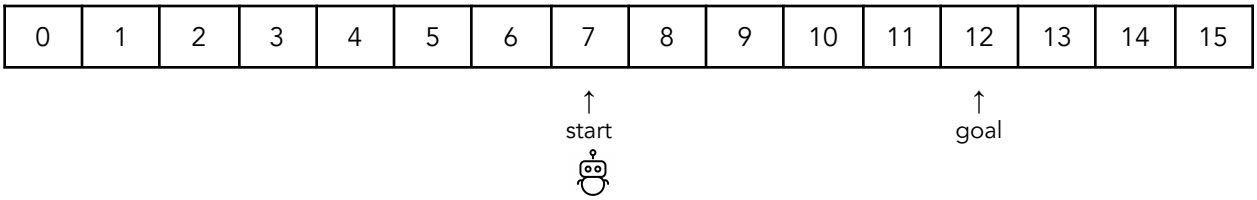
Assignment 5 - Synthesis

For this assignment, you'll develop a synthesizer in Z3.

In class, we played around with a Z3 program for synthesizing sequences of simple robot motion planning primitives in a 1D environment. Example:



Our in-class robot could move left or right.

Task 1

For this assignment, extend the in-class synthesizer so that the robot has a 2D environment and can move left, right, up, or down. How will you represent the environment?

Examples:

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

0,0	1,0	2,0	3,0
0,1	1,1	2,1	3,1
0,2	1,2	2,2	3,2
0,3	1,3	2,3	3,3

The coordinate system on the right is the coordinate system we use for communicating start location and goal location. This will matter for the autograder—for example, if the start location is (0,0) and the end location is (1,1), the autograder will require that the synthesized program moves the robot one square down and one square right, although not necessarily in that order. Internally, you should feel free to use any representation of the environment that helps your implementation.

Our in-class robot used 0 to represent a left instruction and 1 for a right instruction. For this assignment, please use 0 for left, 1 for right, 2 for down, and 3 for up. Again, this will matter for the autograder.

If the robot tries to run an instruction that would move it out of the grid, it stays in the original square instead.

Task 2

Implement iterative deepening. Your program should first try searching for instruction sequences of length 1, then length 2, and so on.

Task 3

Add obstacles to your environment. When the robot runs an instruction that would move it into a square with an obstacle, it should stay in its original position instead. How will you represent obstacles?

The autograder will use the same coordinate system for inputting obstacles that it uses for start and goal location inputs.

Hint: Remember to build up the constraints on the robot's movements using Z3 constraints instead of normal Python. E.g., You'll want to use `And(x,y)` instead of `x and y`. You'll want to use a bunch of Ands instead of if val in list.

Some Useful Resources
https://ericpony.github.io/z3py-tutorial/guide-examples.htm
https://z3prover.github.io/api/html/namespacez3py.html
Some Fun Resources
https://www.cs.cornell.edu/~asampson/blog/minisynth.html
https://www.mattkeeter.com/projects/synthesis/