

31st Annual

Rowan University Programming Contest

hosted by the

Computer Science Department

Friday, 7 April 2017



31st Annual

Rowan University Programming Contest

hosted by the

Computer Science Department

Friday, 7 April 2017

Simulated Terrain Navigation



Abstracted Terrain Map

In order for a robot to move about, it needs an internal representation of the terrain. Ours is a two-dimensional array of numbers:

2	0	0	0	-1
3	0	0	-1	-2
1	0	0	-1	-2
0	0	0	0	-1
0	0	0	0	-1

This map, for example, might represent a flat area with a ridge to the left and a stream to the right.

The robot's location is marked with a red box.

Robot Capabilities

In order for a robot to move about, it has to know its limits.

The robot's **strength** is its ability to go over or remove obstacles, which will be represented on the map by positive numbers.

The robot's **agility** is its ability to navigate ditches or holes, which will be represented on the map by negative numbers.

Robot Instructions

The robot can move in any of four directions, North, South, East, and West, which will be represented by the letters {N, S, E, W}.

Robot Actions

When a robot is instructed to move, it has to check the location first.

Obstacles (positive numbers)

If the location has a zero, or a positive number less than the robot's strength, it moves to that square.

If the location has a positive number equal to the robot's strength, the target location is set to zero and the robot moves onto it.

If the location has a positive number greater than the robot's strength, the robot does not move, but the number in the target location is lowered by the robot's strength value.

Robot Actions

When a robot is instructed to move, it has to check the location first.

Ditches or holes (negative numbers)

If the location has a negative number whose absolute value is less than or equal to the robot's agility, it moves to that square.

If the location has a negative number whose absolute value is greater than the robot's agility, the robot does nothing.

Robot Actions

When a robot is instructed to move, it has to check the location first.

Edge of the Map

If the location is outside the mapped area, such as rolling off the table in the lab, the robot should print a warning message and stop processing instructions.

Robot Actions

In the starting position on the left, the robot has been instructed to go North, the location has a 0, so the robot moves, giving the position on the right.

2	0	0	0	-1
3	0	0	-1	-2
1	0	0	-1	-2
0	0	0	0	-1
0	0	0	0	-1

2	0	0	0	-1
3	0	0	-1	-2
1	0	0	-1	-2
0	0	0	0	-1
0	0	0	0	-1

Robot Actions

Assume a robot with strength 2 is in position on the left, and has been instructed to go West. That location has a 1 so the robot moves, giving the position on the right.

2	0	0	0	-1
3	0	0	-1	-2
1	0	0	-1	-2
0	0	0	0	-1
0	0	0	0	-1

2	0	0	0	-1
3	0	0	-1	-2
1	0	0	-1	-2
0	0	0	0	-1
0	0	0	0	-1

Robot Actions

Assume a robot with strength 2 is in position on the left, and has been instructed to go North. That location has a 3 so the robot cannot move, but it reduces the value of the obstacle by its strength value.

2	0	0	0	-1
3	0	0	-1	-2
1	0	0	-1	-2
0	0	0	0	-1
0	0	0	0	-1

2	0	0	0	-1
1	0	0	-1	-2
1	0	0	-1	-2
0	0	0	0	-1
0	0	0	0	-1

Robot Actions

Assume a robot with agility 1 is in position on the left, and has been instructed to go East. That location has a -1, so the robot moves to that location.

2	0	0	0	-1
3	0	0	-1	-2
1	0	0	-1	-2
0	0	0	0	-1
0	0	0	0	-1

2	0	0	0	-1
1	0	0	-1	-2
1	0	0	-1	-2
0	0	0	0	-1
0	0	0	0	-1

Robot Actions

Assume a robot with agility 1 is in position on the left, and has been instructed to go East. That location has a -2, so the robot does nothing.

2	0	0	0	-1
3	0	0	-1	-2
1	0	0	-1	-2
0	0	0	0	-1
0	0	0	0	-1

2	0	0	0	-1
1	0	0	-1	-2
1	0	0	-1	-2
0	0	0	0	-1
0	0	0	0	-1

Robot Actions

Assume a robot is in the position on the left, and has been instructed to go South. That location is off the grid, so the robot quits.

2	0	0	0	-1
3	0	0	-1	-2
1	0	0	-1	-2
0	0	0	0	-1
0	0	0	0	-1

2	0	0	0	-1
1	0	0	-1	-2
1	0	0	-1	-2
0	0	0	0	-1
0	0	0	0	-1

Robot Actions

Write a program which reads in a grid, the strength and agility of a robot, the robot's starting position on the grid, and the list of instructions the robot is to process.

After processing the instructions, print the robot's location and the values in the grid.