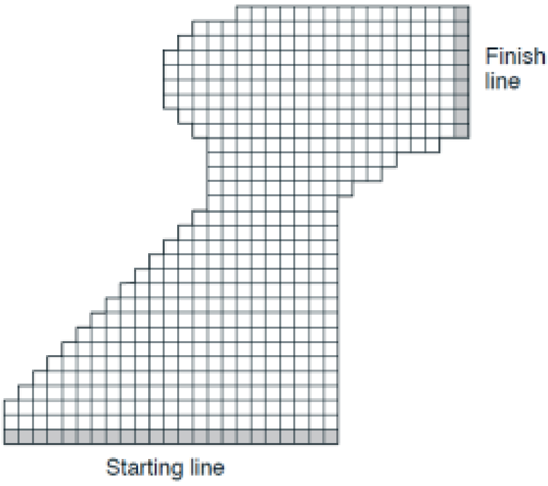
Consider driving a race car around a turn like those shown in Figure.



You want to go as fast as possible, but not so fast as to run off the track. In our simplified racetrack, the car is at one of a discrete set of grid positions, the cells in the diagram. The velocity is also discrete, a number of grid cells moved horizontally and vertically per time step. The actions are increments to the velocity components. Each may be changed by +1, -1, or 0 in one step, for a total of nine actions. Both velocity components are restricted to be nonnegative and less than 5, and they cannot both be zero. Each episode begins in one of the randomly selected start states and ends when the car crosses the finish line. The rewards are -1 for each step that stays on the track, and -5 if the agent tries to drive off the track. When the car goes off track, it will start from an initial random starting position.

Solve this problem using Q(Lambda), SARSA(Lambda). Try Lambdas (0.00, 0.95, 1.00). Set alpha to 0.1 and the discount rate to 1. You can end the episode at 200 if the car was unable to reach the finish line. Run the algorithms as long as they provide reasonable results. Report the optimal policies and values for each state. You can simulate an optimal policy starting from starting line with velocity 0 and observe the track.