1. Shortest Path in a Graph as an MDP Consider the problem of finding the shortest path in a graph \((V, E)\) (with \(V\) the set of vertices, and \(E\) the set of directed edges), and where \(v_g\) is the destination vertex. Every edge in the graph takes equally long to traverse. Describe how this problem can be encoded into a Markov decision process, \((S, A, T, \gamma, R)\), such that the optimal solution in the Markov decision process is the shortest path in the graph. Your MDP is required to have rewards that are positive (zero included) for all transitions.

2. (a) Implement value iteration into value iteration.m. Run value iteration for the provided gridworld and report the values obtained for each state. (You can check your results against the ones provided in the starter-code to ensure correctness!) Look at main.m to get started. The starter code is available at the webpage cs685/hw4.mdp.tar

(b) Queuing. Consider a server that has to serve three queues denoted by a, b, c. Each queue can be of length zero through five. At the beginning of each time-step, the server gets to choose between each of the three queues. If the server stays with its current queue and the current queue is non-empty, then it clears a request from that queue. If the server moves to another queue, this takes up an entire time-step and no request is cleared in that time-step. Clearing a request results in a reward of +1, moving between queues results in a reward of 0. In each time-step, after the server has chosen its action and this action has been executed, a request is added to each queue with size strictly smaller than five with probability \(p_a, p_b, p_c\) respectively. This happens independently for each of the queues. The discount factor \(\gamma = 0.99\). (i) Implement the queue MDP, \((S, A, T, \gamma, R)\), that models this setting. (ii) For \(p_a = 0.1, p_b = 0.2, p_c = 0.5, \gamma = 0\) using value iteration (which you already implemented for part (a)) compute the values and optimal actions for all states. Report the values and the optimal actions for the following states: (0, 0, 0, a), (2, 0, 4, b), (5, 2, 1, c).