8.2 Value and policy iteration

In this problem, you will use value and policy iteration to find the optimal policy of the MDP demonstrated in class. This MDP has $|\mathcal{S}| = 81$ states, $|\mathcal{A}| = 4$ actions, and discount factor $\gamma = 0.9925$. Download the ASCII files on the course web site that store the transition matrices and reward function for this MDP. The transition matrices are stored in a sparse format, listing only the row and column indices with non-zero values; if loaded correctly, the rows of these matrices should sum to one.

(a) Compute the optimal state value function $V^*(s)$ using the method of value iteration. Print out a list of the non-zero values of $V^*(s)$. Compare your answer to the numbered maze shown below. The correct value function will have positive values at all the numbered squares and negative values at the all squares with dragons.

(b) Compute the optimal policy $\pi^*(s)$ from your answer in part (a). Interpret the four actions in this MDP as (probable) moves to the WEST, NORTH, EAST, and SOUTH. Fill in the correspondingly numbered squares of the maze with arrows that point in the directions prescribed by the optimal policy. Turn in a copy of your solution for the optimal policy, as visualized in this way.

(c) Compute the optimal policy $\pi^*(s)$ using the method of policy iteration. For the numbered squares in the maze, does it agree with your result from part (b)? (It should.) Use this check to make sure that your answers in part (a) are correct to at least two decimal places. Also answer the following questions: (i) if you start with the initial policy that points EAST in every state, how many iterations are required for convergence? (ii) if you start with the initial policy that points SOUTH in every state, how many iterations are required for convergence?

(d) **Turn in your source code for all the above questions.** As usual, you may program in the language of your choice.