∞, $|A| < \infty$). In this section, we will also assume that we know the state transition probabilities $\{P_{sa}\}$ and the reward function R.

The first algorithm, **value iteration**, is as follows:

Algorithm 1 Value Iteration

1: For each state s, initialize $V(s) := 0$.
2: **for** until convergence **do**
3: For every state, update

$$V(s) := R(s) + \max_{a \in A} \gamma \sum_{s'} P_{sa}(s')V(s'). \tag{4}$$

This algorithm can be thought of as repeatedly trying to update the estimated value function using Bellman Equations (2).

There are two possible ways of performing the updates in the inner loop of the algorithm. In the first, we can first compute the new values for $V(s)$ for every state s, and then overwrite all the old values with the new values. This is called a **synchronous** update. In this case, the algorithm can be viewed as implementing a "Bellman backup operator" that takes a current estimate of the value function, and maps it to a new estimate. (See homework problem for details.) Alternatively, we can also perform **asynchronous** updates. Here, we would loop over the states (in some order), updating the values one at a time.

Under either synchronous or asynchronous updates, it can be shown that value iteration will cause V to converge to V^*. Having found V^*, we can then use Equation (3) to find the optimal policy.

Apart from value iteration, there is a second standard algorithm for finding an optimal policy for an MDP. The **policy iteration** algorithm proceeds as follows:

Algorithm 2 Policy Iteration

1: Initialize π randomly.
2: **for** until convergence **do**
3: Let $V := V^\pi$. ▷ typically by linear system solver
4: For each state s, let

$$\pi(s) := \arg\max_{a \in A} \sum_{s'} P_{sa}(s')V(s').$$