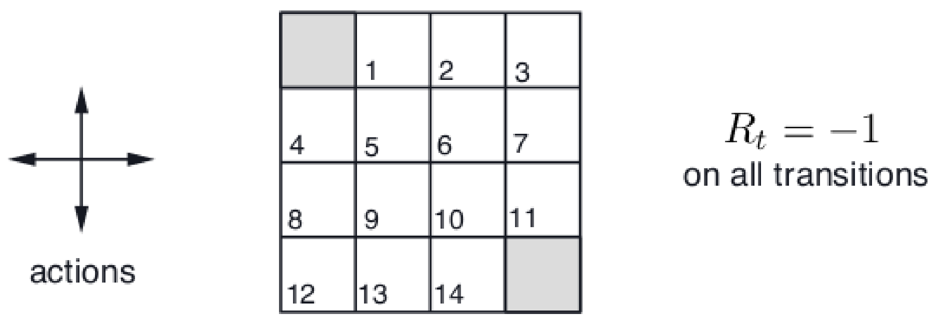
Worksheet 5

1. In iterative policy evaluation, we seek to find the value function for a policy π by applying the Bellman equation many times to generate a sequence of value functions v_k that will eventually converge to the true value function v_π. How can we modify the update below to generate a sequence of action value functions q_k?

$$v_{k+1}(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a) \left[r + \gamma v_k(s')\right]$$

2. A deterministic policy $\pi(s)$ outputs an action $a \in \mathcal{A} = \{a_1, a_2, \ldots, a_k\}$ directly. More generally, a policy $\pi(\cdot|s)$ outputs the probabilities for all actions: $\pi(\cdot|s) = [\pi(a_1|s), \pi(a_2|s), \ldots, \pi(a_k|s)$. How can you write a deterministic policy in this form? Let $\pi(s) = a_i$ and define $\pi(\cdot|s)$.

3. (*Exercise 4.1 S&B)* Consider the 4x4 gridworld below, where actions that would take the agent off the grid leave the state unchanged. The task is episodic with $\gamma = 1$ and the terminal states are the shaded blocks. Using the precomputed values for the equiprobable policy below, what is $q_\pi(11, \text{down})$? What is $q_\pi(7, \text{down})$?



$k = \infty$

0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0

4. (*Exercise 4.1 from S&B*) Suppose in the above gridworld where a new state 15 is added to the gridworld just below state 13, and its actions, left, up, right, and dowm, take the agent to the states 12, 13, 14, and 15, respectively. Assume that the transitions from the original states are unchanged. What, then is, $v_\pi(15)$ for the equiprobable random policy? Now suppose the dynamics of state 13 are also changed, such that action down from state 13 takes the agent to the new state 15. What is $v_\pi(15)$ for the equiprobable random policy in this case?

5. **(Challenge Question)** A gambler has the opportunity to make bets on the outcomes of a sequence of coin flips. If the coin comes up heads, she wins as many dollars as she has staked on that flip; if it is tails, she loses her stake. The game ends when the gambler wins by reaching her goal of \$100, or loses by running out of money. On each flip, the gambler must decide what portion of her capital to stake, in integer numbers of dollars. This problem can be formulated as an undiscounted, episodic, finite MDP. The state is the gambler's capital, $s \in \{1, 2, ..., 99\}$ and the actions are stakes, $a \in \{0, 1, ..., \min(s, 100 - s)\}$. The reward is $+1$ when reaching the goal of \$100 and zero on all other transitions. The probability of seeing heads is $p_h = 0.4$.

(a) What does the value of a state mean in this problem? For example, in a gridworld where the value of 1 per step, the value represents the expected number of steps to goal. What does the value of state mean in the gambler's problem? Think about the minimum and maximum possible values, and think about the values of state 50 (which is 0.4) and state 99 (which is near 0.95).

(b) Modify the pseudocode for value iteration to more efficiently solve this specific problem, by exploiting your knowledge of the dynamics. *Hint: Not all states transition to every other state. For example, can you transition from state* 1 *to state* 99*?*

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

```
Loop:
|  Δ ← 0
|  Loop for each s ∈ S:
|       v ← V(s)
|       V(s) ← max_a Σ_{s',r} p(s',r|s,a)[r + γV(s')]
|       Δ ← max(Δ, |v − V(s)|)
until Δ < θ
```

Output a deterministic policy, $\pi \approx \pi_*$, such that
$\pi(s) = \operatorname{argmax}_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$

Worksheet 5

6. **(Challenge Question)** (*Exercise 4.4 S&B*) The policy iteration algorithm on page 80 has a subtle bug in that it may never terminate if the policy continually switches between two or more policies that are equally good. This is ok for pedagogy, but not for actual use. Modify the pseudocode so that convergence is guaranteed. Note that there is more than one approach to solve this problem.

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
   $\quad \Delta \leftarrow 0$
   $\quad$ Loop for each $s \in \mathcal{S}$:
   $\quad\quad v \leftarrow V(s)$
   $\quad\quad V(s) \leftarrow \sum_{s',r} p(s', r \,|\, s, \pi(s))\big[r + \gamma V(s')\big]$
   $\quad\quad \Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   $\quad$ *old-action* $\leftarrow \pi(s)$
   $\quad \pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r \,|\, s, a)\big[r + \gamma V(s')\big]$
   $\quad$ If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2