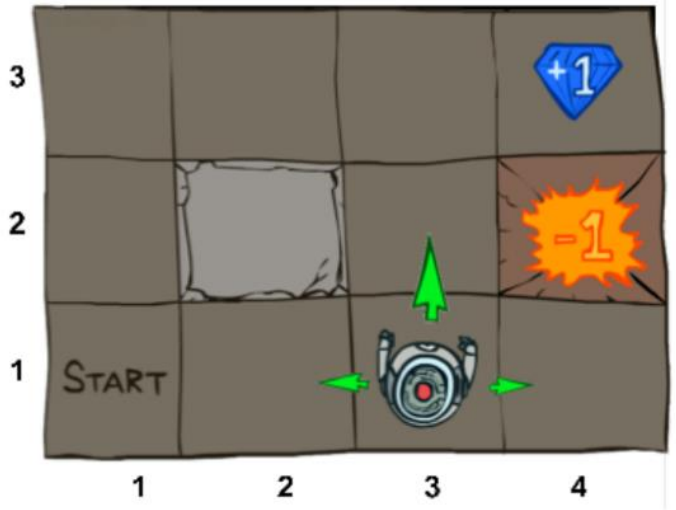
Value iteration

$\gamma = 1$, Finite horizon: T=100

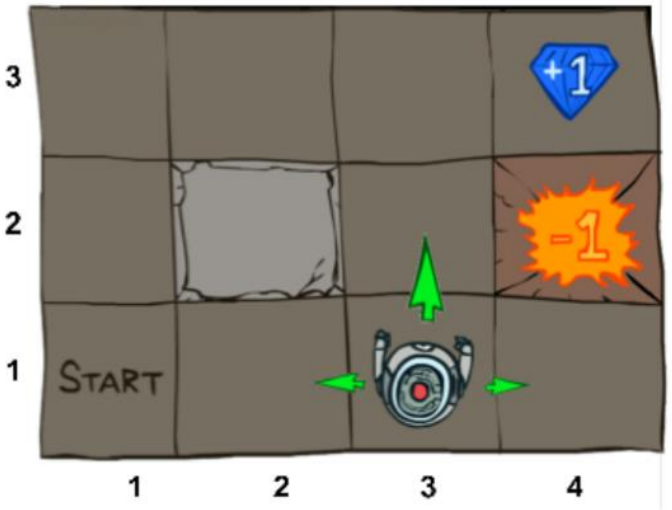
$\gamma = 1$, Finite horizon: T=100

- $V_{n+1}(s) = \max_{a \in A}\{r(s,a) + \gamma \sum_{\{s' \in S\}} p(s'|s,a) V_n(s')\}, s \in S$
- V*(3, 4)=
- V*(2,4)=
- V*(i,3)=
- V*(i,2)=
- V*(i,1)=
- V*(1,4)=



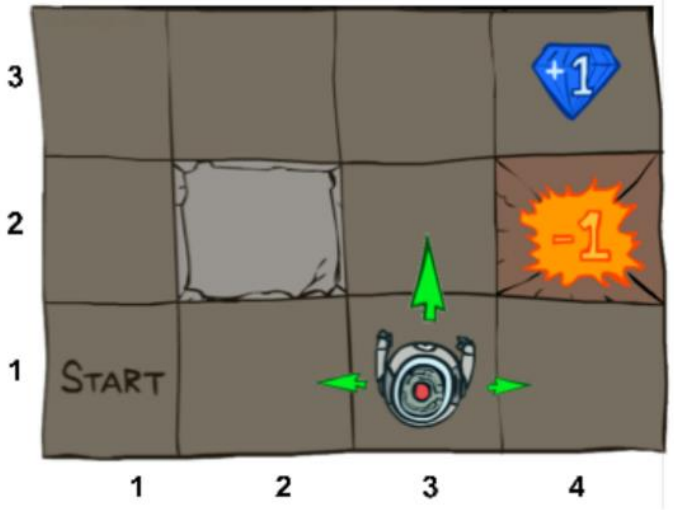
$\gamma = 1$, Finite horizon: T=100

- $V_{n+1}(s) = \max_{a \in A} \{ r(s,a) + \gamma \sum_{\{s' \in S\}} p(s'|s,a) V_n(s') \}, s \in S$
- V*(3, 4)= 1
- V*(2,4)= -1
- V*(i,3)=
- V*(i,2)=
- V*(i,1)=
- V*(1,4)=



$\gamma = 1$, Finite horizon: T=100

- $V_{n+1}(s) = \max_{a \in A} \{ r(s,a) + \gamma \sum_{\{s' \in S\}} p(s'|s,a) V_n(s') \}, s \in S$
- V*(3, 4)= 1
- V*(2,4)= -1
- V*(i,3)= 1
- V*(i,2)= 1
- V*(i,1)= 1
- V*(1,4)= 1

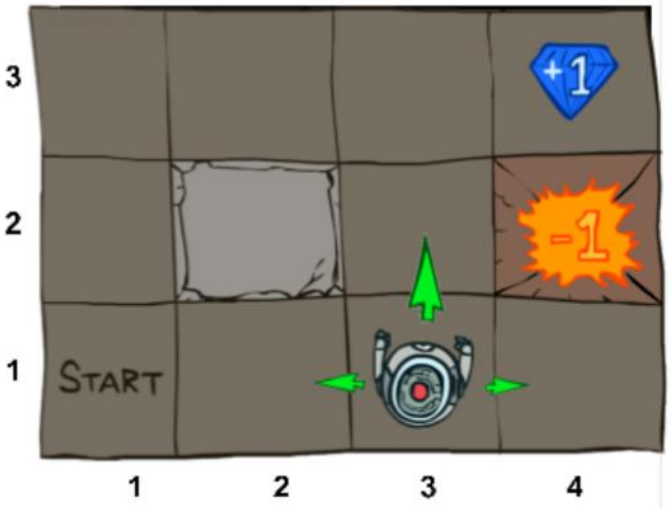


Value iteration

$\gamma = 0.9$, Finite horizon: T=100

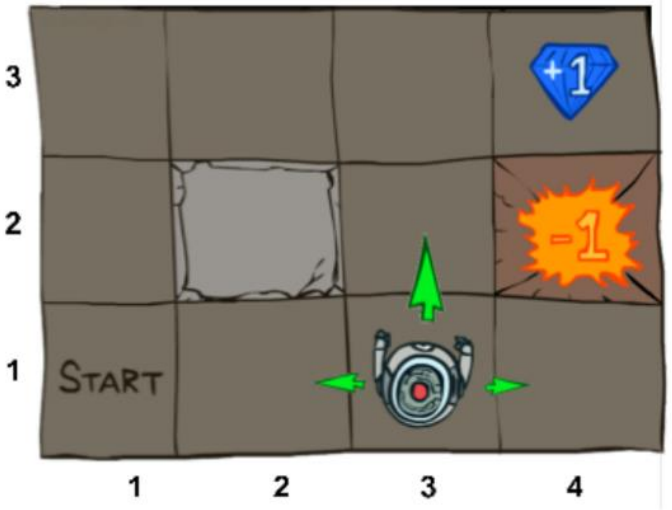
$\gamma = 0.9$, Finite horizon: T=100

- $V_{n+1}(s) = \max_{a \in A} \{ r(s,a) + \gamma \sum_{\{s' \in S\}} p(s'|s,a) V_n(s') \}, s \in S$
- V*(3, 4)=
- V*(2,4)=
- V*(3,3)=
- V*(3,2)=
- V*(1,1)=



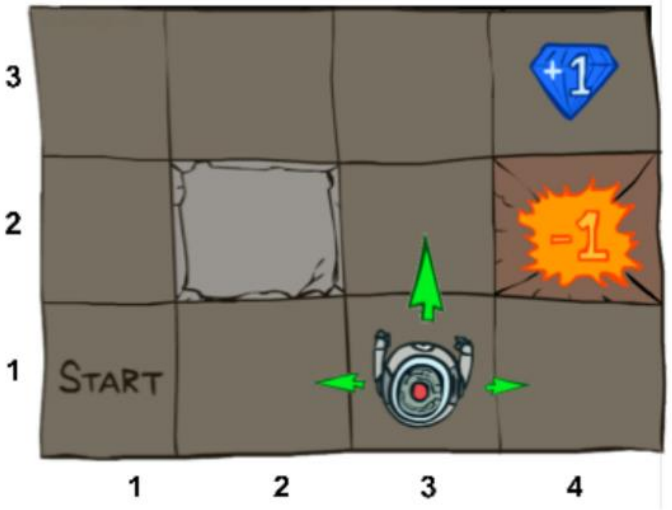
$\gamma = 0.9$, Finite horizon: T=100

- $V_{n+1}(s) = \max_{a \in A} \{ r(s,a) + \gamma \sum_{\{s' \in S\}} p(s'|s,a) V_n(s') \}, s \in S$
- V*(3, 4)= 1
- V*(2,4)= -1
- V*(3,3)=
- V*(3,2)=
- V*(1,1)=



$\gamma = 0.9$, Finite horizon: T=100

- $V_{n+1}(s) = \max_{a \in A}\{r(s,a) + \gamma \sum_{\{s' \in S\}} p(s'|s,a) V_n(s')\}, s \in S$
- V*(3, 4)= 1
- V*(2,4)= -1
- V*(3,3)= 0.9
- V*(3,2)= 0.9*0.9=0.81
- V*(1,1)= 0.9*0.9*0.9*0.9*0.9=0.59
- …



Value iteration

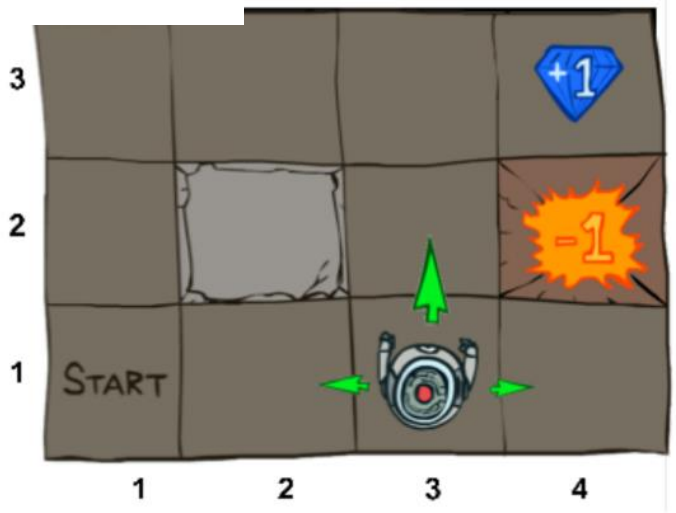
- $\gamma = 0.9$, **Infinite horizon**Transition matrix:
- With probability 0.8, the requested action is preformed.
- With probability 0.1, the robot gets to the square which is **left** of the requested direction.
- With probability 0.1, the robot gets to the square which is **right** of the requested direction.

$V_0^*(s) = \text{optimal value for state s when } k = 0$
$V_0^*(s) = 0, \forall s$

$V_1^*(s) = \text{optimal value for state s when } k = 1$
$V_1^*(s) = \max_{a \in A} \sum_{s'} p(s' \mid s, a)(r(s, a, s') + \gamma V_0^*(s'))$

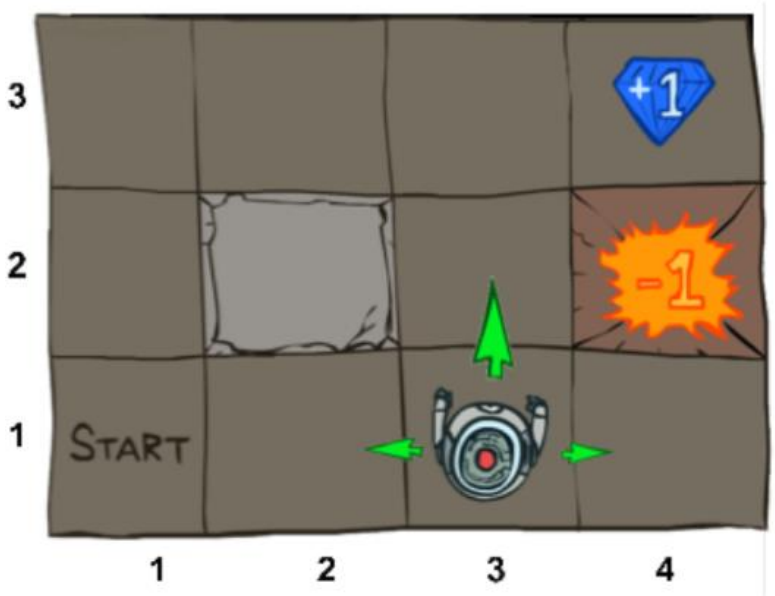
$V_2^*(s) = \text{optimal value for state s when } k = 2$
$V_2^*(s) = \max_{a \in A} \sum_{s'} p(s' \mid s, a)(r(s, a, s') + \gamma V_1^*(s'))$

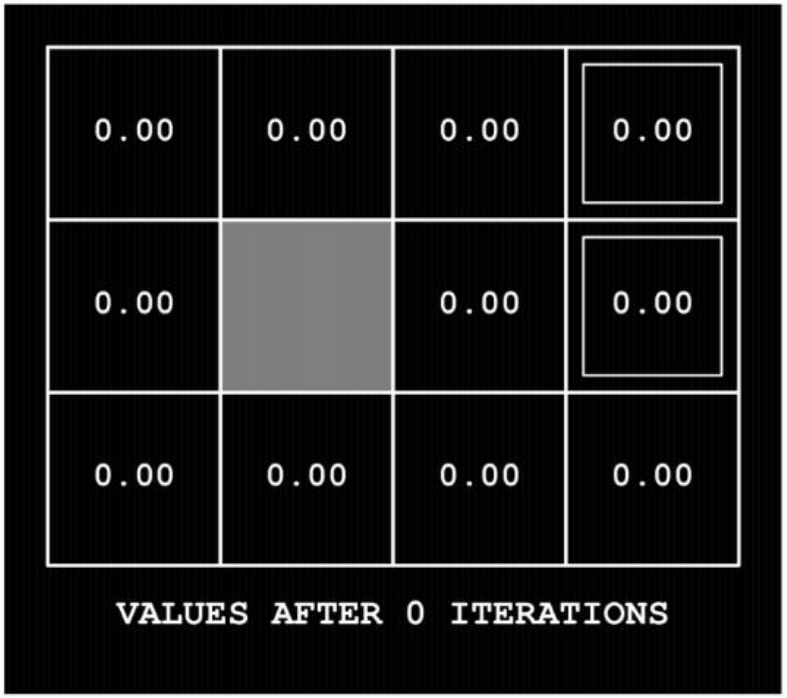
$V_k^*(s) = \text{optimal value for state s, for iteration } k$
$V_k^*(s) = \max_{a \in A} \sum_{s'} p(s' \mid s, a)(r(s, a, s') + \gamma V_{k-1}^*(s'))$



$\gamma = 0.9$, noise=0.2- $k = 0$

- $V_0(s) = 0$





$\gamma = 0.9$, noise=0.2- $k = 1$

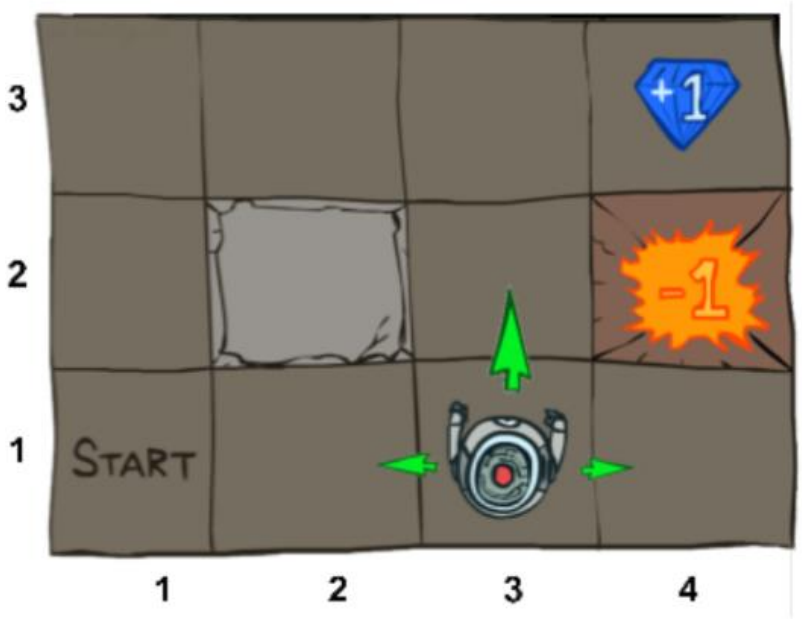
- $V_1(s) = \max_{a \in A} \{ r(s,a) + \gamma \sum_{\{s' \in S\}} p(s'|s,a) V_0(s') \}, s \in S$

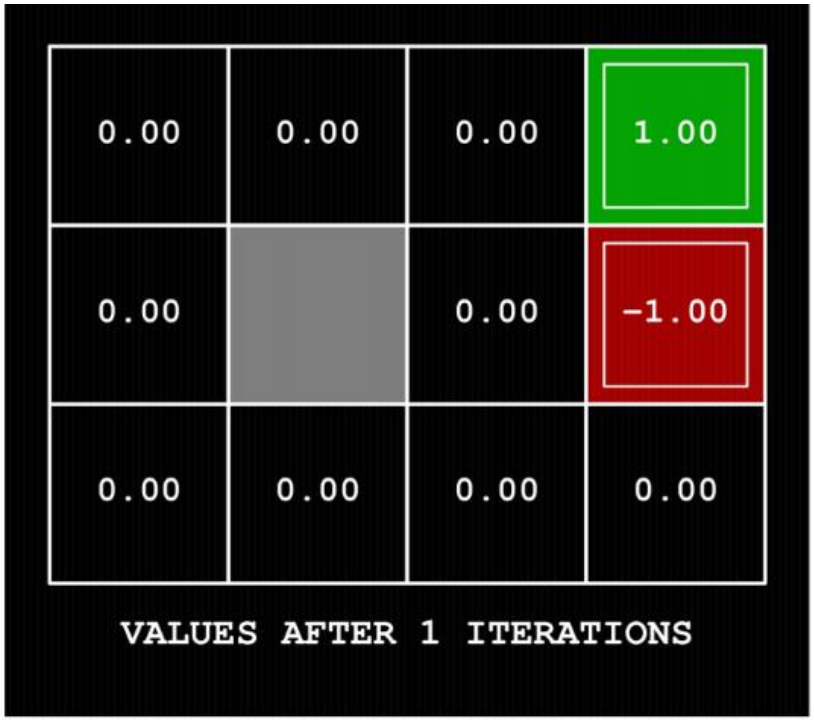
for $k = 1$:

$V_1(3,4) = 1$

$V_1(2,4) = -1$

$V_1(s) = \max_{a \in \{u,l,r\}} (0 + \sum_{s'} p(s' \mid s,a) V_0^*(s')) = 0, s \notin (2,4), (3,4)$





$\gamma = 0.9$, noise=0.2- $k = 2$

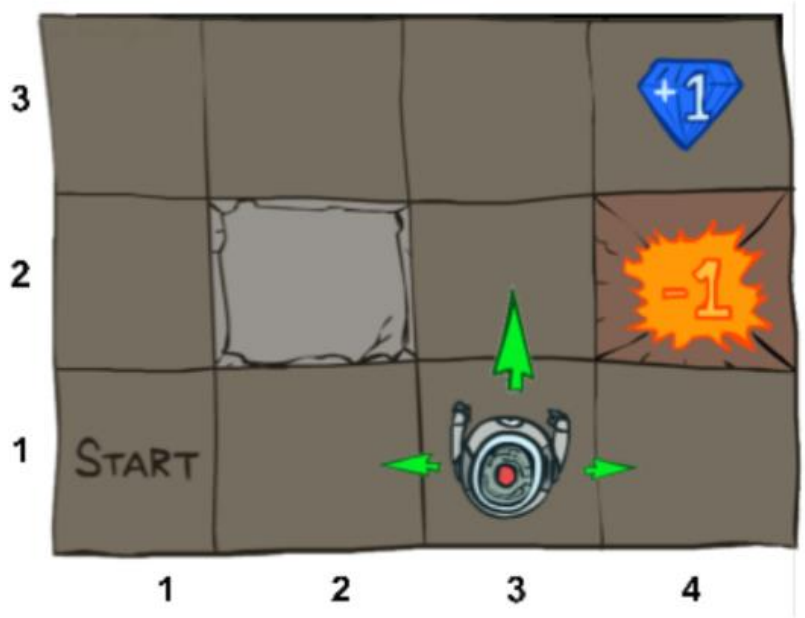
for $k = 2$:

$V_2(3,4) = 1$

$V_2(2,4) = -1$

$V_2(3,3) = \max_{a \in A} \sum_{s'} p(s' \mid s,a)(r(s,a,s') + \gamma V_1^*(s')) = 0.8(0 + 0.9 * 1) = 0.72$

$V_2(s) = \max_{a \in A} \sum_{s'} p(s' \mid s,a)(r(s,a,s') + \gamma V_1^*(s')) = 0.8(0 + 0.9 * 0) = 0, s \notin (2,4), (3,4), (3,3)$



$\gamma = 0.9$, noise=0.2- $k = 3$

for $k = 3$:

$V_3(3,4) = 1$

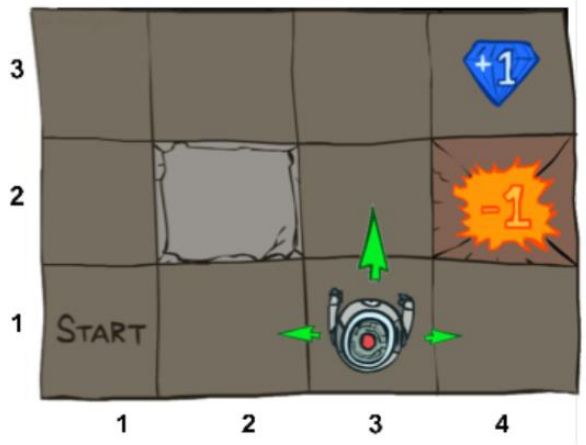
$V_3(2,4) = -1$

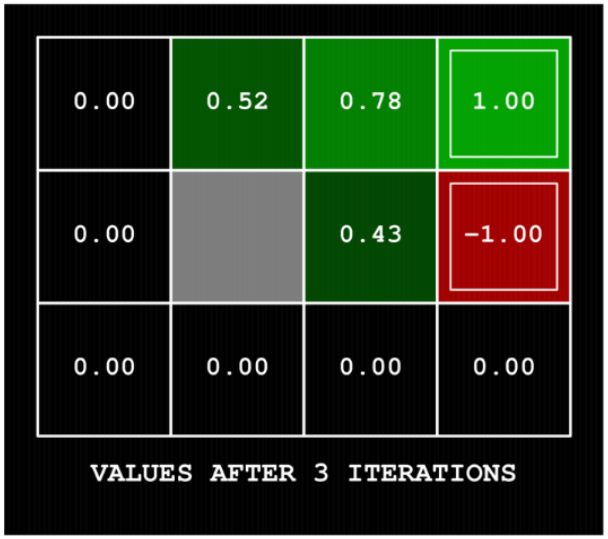
$V_3(3,3) = \max_{a \in A} \sum_{s'} p(s' \mid s,a)(r(s,a,s') + \gamma V_2^*(s')) = 0.8(0 + 0.9 * 1) + 0.1(0 + 0.9 * 0.72) + 0.1(0 + 0.9 * 0) = 0.78$

$V_3(2,3) = \max_{a \in A} \sum_{s'} p(s' \mid s,a)(r(s,a,s') + \gamma V_2^*(s')) = 0.8(0 + 0.9 * 0.72) + 0.1(0 + 0.9 * (-1)) = 0.43$

$V_3(3,2) = \max_{a \in A} \sum_{s'} p(s' \mid s,a)(r(s,a,s') + \gamma V_2^*(s')) = 0.8(0 + 0.9 * 0.72) = 0.52$

$V_2(s) = 0, s \notin (2,4), (3,4), (3,3), (2,3), (3,2)$



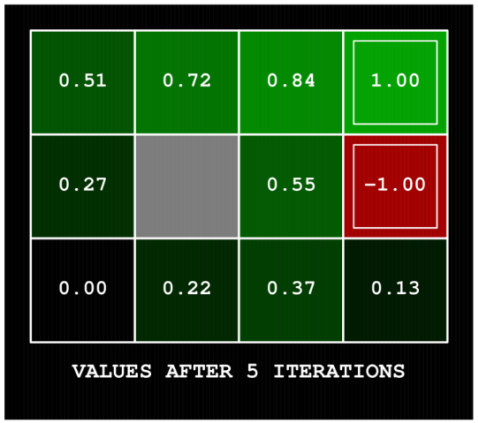


$\gamma = 0.9$, noise=0.2- $k > 3$

- $V_{k+1}(s) = \max_{a \in A} \{ r(s,a) + \gamma \sum_{\{s' \in S\}} p(s'|s,a) V_k(s') \}, s \in S$

$\gamma = 0.9$, noise=0.2- $k > 3$

- $V_{k+1}(s) = \max_{a \in A} \{ r(s,a) + \gamma \sum_{\{s' \in S\}} p(s'|s,a) V_k(s') \}, s \in S$



$\gamma = 0.9$, noise=0.2- $k > 3$

- $V_{k+1}(s) = \max_{a \in A} \{ r(s,a) + \gamma \sum_{\{s' \in S\}} p(s'|s,a) V_k(s') \}, s \in S$

$\gamma = 0.9$, noise=0.2- $k > 3$

- $V_{k+1}(s) = \max_{a \in A} \{ r(s,a) + \gamma \sum_{\{s' \in S\}} p(s'|s,a) V_k(s') \}, s \in S$

$\gamma = 0.9$, noise=0.2- $k > 3$

- $V_{k+1}(s) = \max_{a \in A} \{ r(s,a) + \gamma \sum_{\{s' \in S\}} p(s'|s,a) V_k(s') \}, s \in S$

$\gamma = 0.9$, noise=0.2- $k > 3$

- $V_{k+1}(s) = \max_{a \in A} \{ r(s,a) + \gamma \sum_{\{s' \in S\}} p(s'|s,a) V_k(s') \}, s \in S$

$\gamma = 0.9$, noise=0.2- $k > 3$

- $V_{k+1}(s) = \max_{a \in A} \{ r(s,a) + \gamma \sum_{\{s' \in S\}} p(s'|s,a) V_k(s') \}, s \in S$

$\gamma = 0.9$, noise=0.2- $k > 3$

- $V_{k+1}(s) = \max_{a \in A} \{ r(s,a) + \gamma \sum_{\{s' \in S\}} p(s'|s,a) V_k(s') \}, s \in S$

$\gamma = 0.9$, noise=0.2- $k > 3$

- $V_{k+1}(s) = \max_{a \in A} \{ r(s,a) + \gamma \sum_{\{s' \in S\}} p(s'|s,a) V_k(s') \}, s \in S$

$\gamma = 0.9$, noise=0.2- $k > 3$

- $V_{k+1}(s)$
  $= \max_{a \in A} \{ r(s,a) + \gamma \sum_{\{s' \in S\}} p(s'|s,a) V_k(s') \}, s \in S$

$\gamma = 0.9$, noise=0.2- $k > 3$

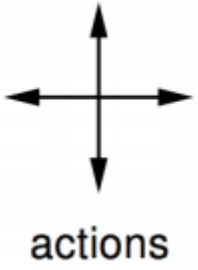
- $V_{k+1}(s)$
  $= \max_{a \in A} \{ r(s,a) + \gamma \sum_{\{s' \in S\}} p(s'|s,a) V_k(s') \}, s \in S$

Policy Iteration

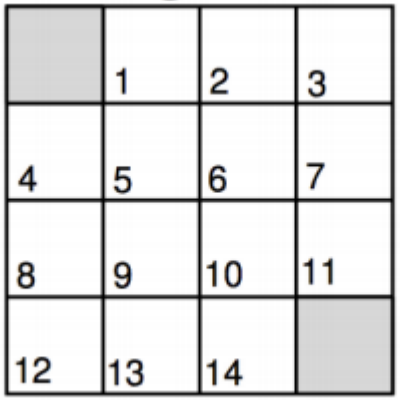
$\gamma = 1$, Finite horizon: T=100

Evaluating a Random Policy in Gridworld

- Undiscounted episodic MDP ($\gamma = 1$)
- Nonterminal states 1, ..., 14
- One terminal state (shown twice as shaded squares)
- Actions leading out of the grid leave state unchanged
- Reward is −1 until the terminal state is reached
- Agent follows uniform random policy



actions



$r = -1$
on all transitions

Solution

- Step 1: $\pi_0(\mathrm{L}|\ \cdot)=\pi_0(\mathrm{R}|\ \cdot)=\pi_0(\mathrm{U}|\ \cdot)=\pi_0(\mathrm{D}|\ \cdot)=0.25$
- Step 2: Policy Evaluation of π_0 :

$k=0$

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0

$k=1$

0.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	0.0

$k=2$

0.0	-1.7	-2.0	-2.0
-1.7	-2.0	-2.0	-2.0
-2.0	-2.0	-2.0	-1.7
-2.0	-2.0	-1.7	0.0

$k=3$

0.0	-2.4	-2.9	-3.0
-2.4	-2.9	-3.0	-2.9
-2.9	-3.0	-2.9	-2.4
-3.0	-2.9	-2.4	0.0

$k=10$

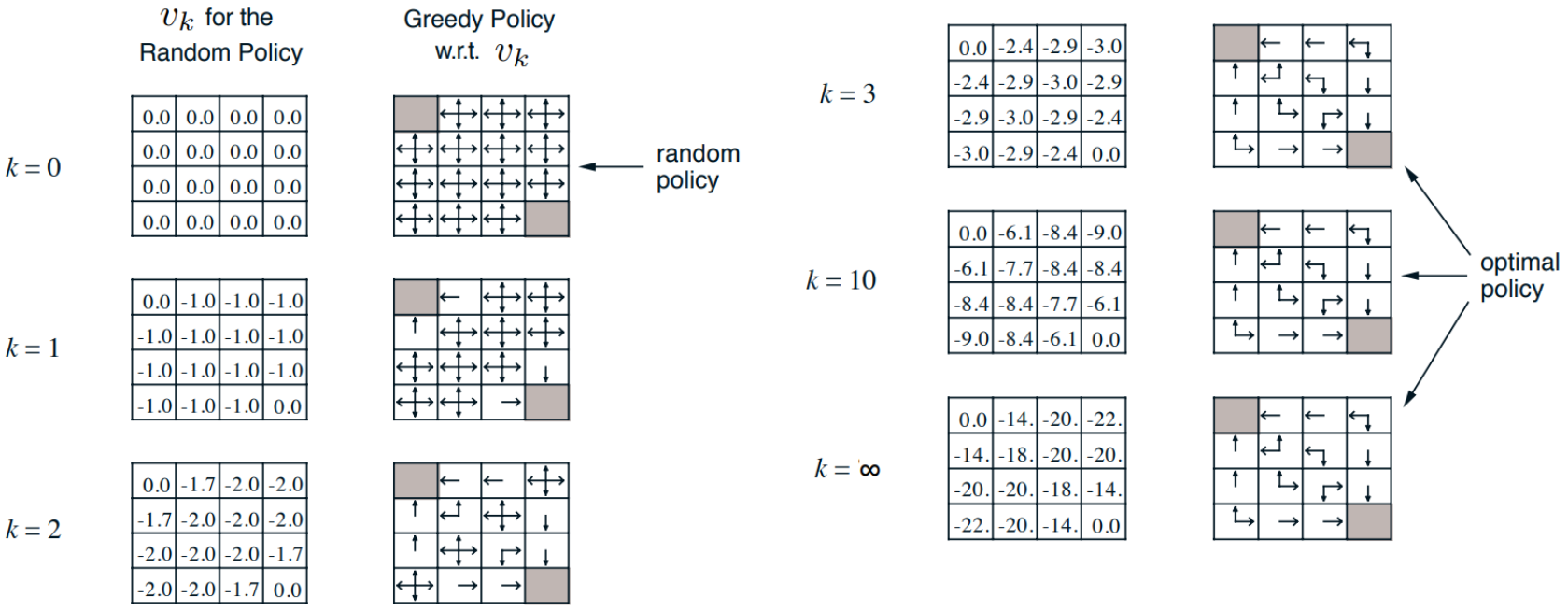
0.0	-6.1	-8.4	-9.0
-6.1	-7.7	-8.4	-8.4
-8.4	-8.4	-7.7	-6.1
-9.0	-8.4	-6.1	0.0

$k=\infty$

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

	1	2	3
4	5	6	7
8	9	10	11
12	13	14	

Step 2: Policy Improvement – Calculate π_1



v_k for the Random Policy

Greedy Policy w.r.t. v_k