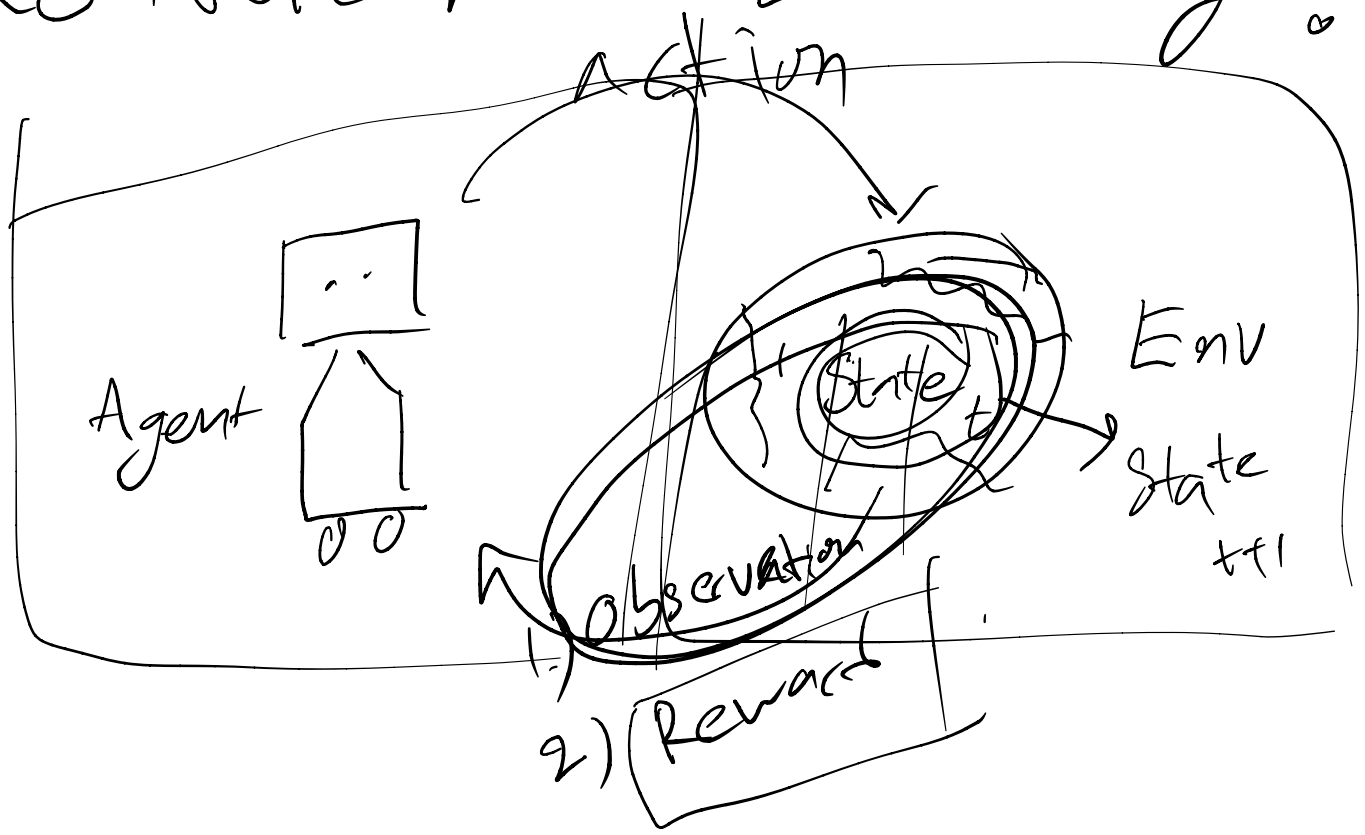
Reinforcement Learning !



What is Environment ?

- Environment maintains a state

$$S = \{ s_1, s_2, s_3, \ldots, s_n \}$$

- State changes

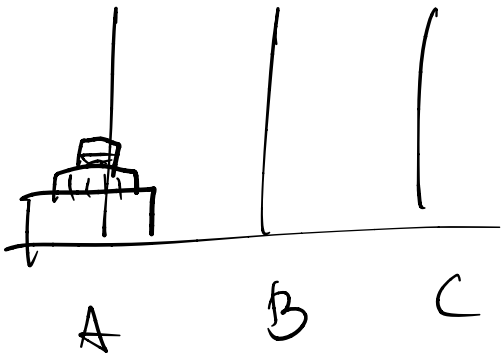
$$s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \ldots$$

- Environment emits reward

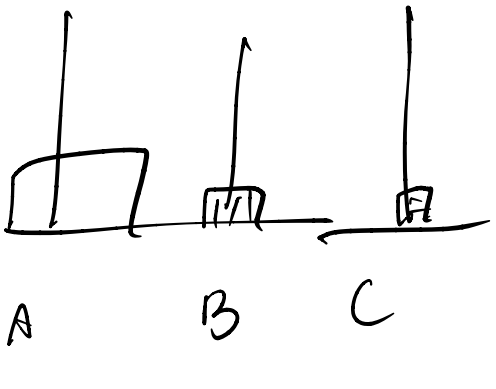
r

Example

Tower of Hanoi



S_1



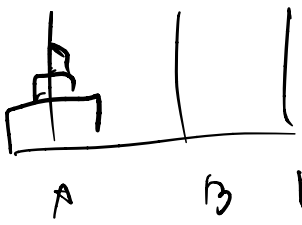
S_2

We can Represent states with

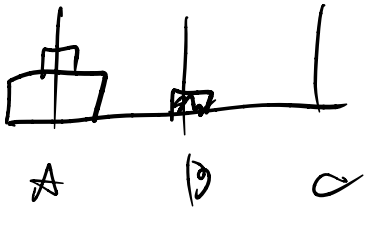
(i) number

(ii) Vector

(iii) Matrix etc.

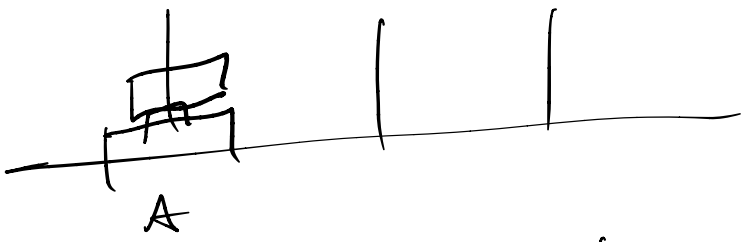


$\rightarrow S = 1$

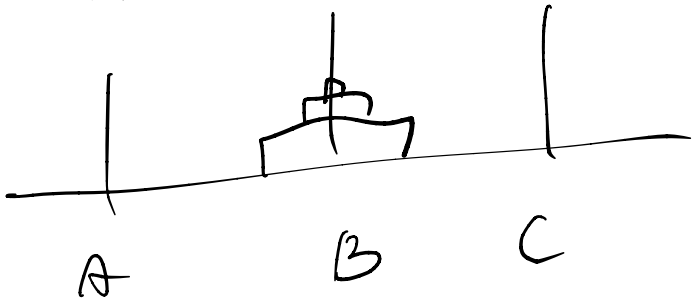


$\rightarrow S = 2$

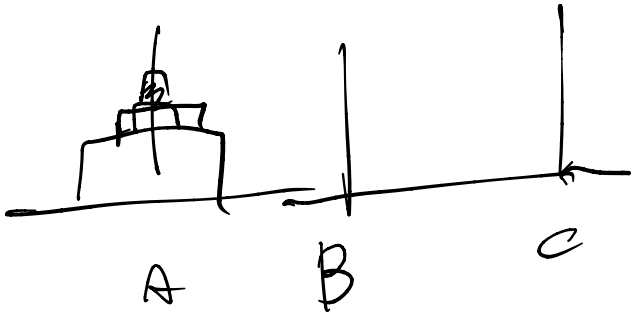
$\rightarrow S = [1 \quad 0 \quad 0]$



$\rightarrow S = [2 \quad 0 \quad 0]$



$\rightarrow S = [0 \quad 1 \quad 0]$



$\rightarrow S = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 0 & 0 \\ 3 & 0 & 0 \end{bmatrix}$

What is Reward ?

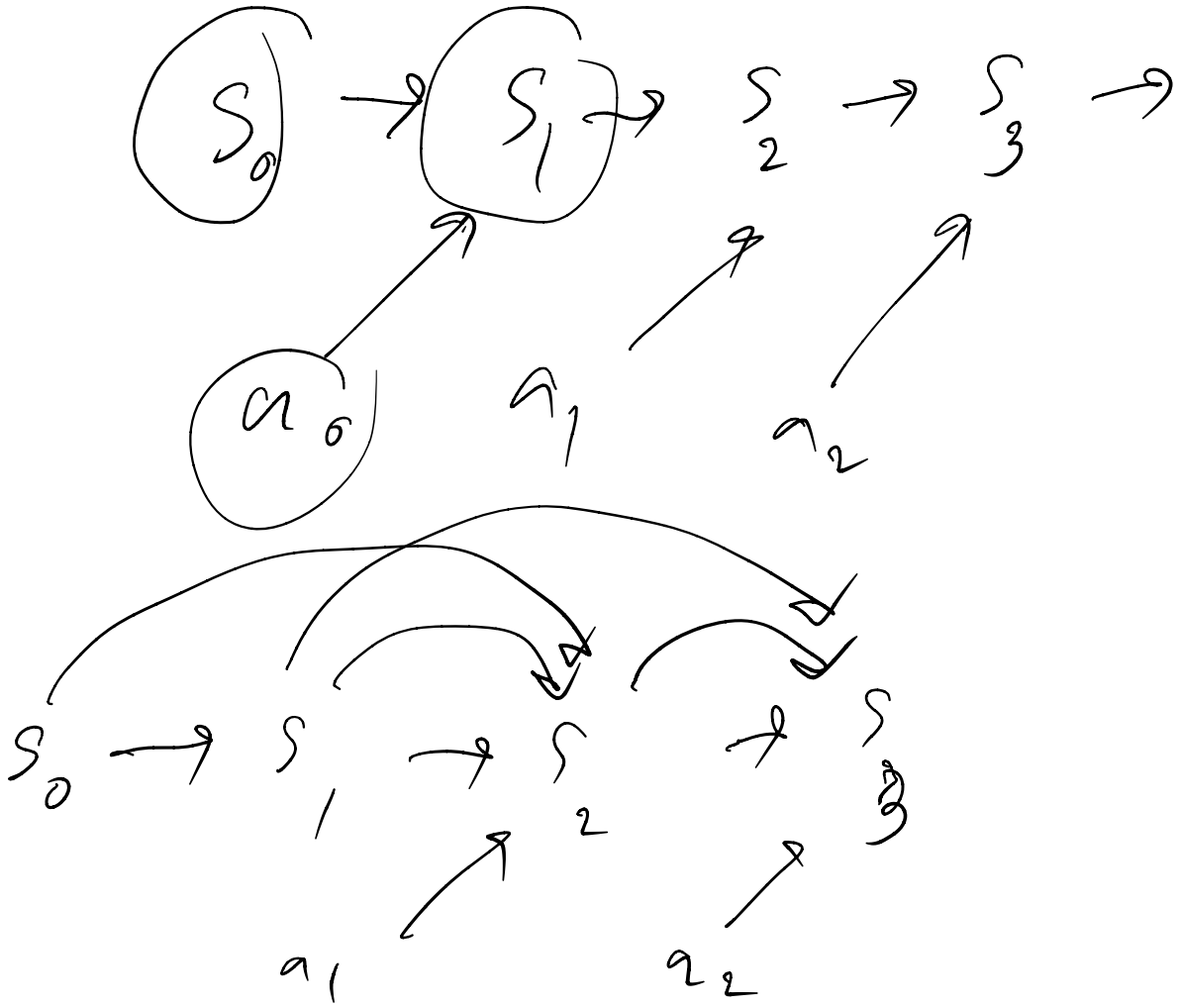
- Reward is a scalar number
- At each time-step, a reward is emitted.

* Reward encapsulates the objective of the task.

What make the state change ?

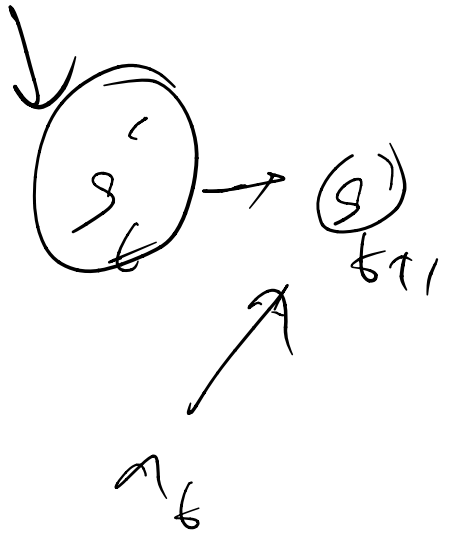
1.) Action

2.) Time



Let's define a transition function

$$T = P\left(S_{t+1} = s' \middle| S_t = s, A_t = a\right)$$



Markov assumption

Markov Decision Process (MDP)

MDP is one way to model environment mathematically.

MDP

$$\langle S, A, T, R, \gamma \rangle$$

$$S = \{ s_0, s_1, s_2, \ldots, s_n \}$$

$$A = \{ a_0, a_1, \ldots, a_m \}$$

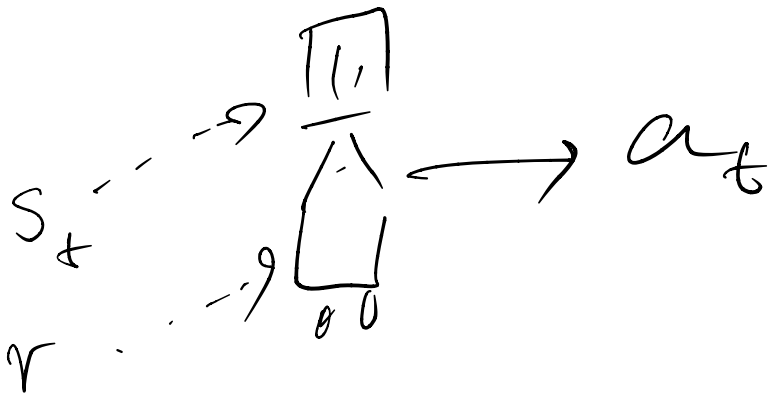
$$T = P(s_{t+1} \mid s_t, a_t)$$

$$R = R(s_{t+1}, s_t, a_t) \rightarrow r$$

γ = discount factor ; $\gamma \in [0, 1]$

$$P(R \mid s_{t+1}, s_t, a_t)$$

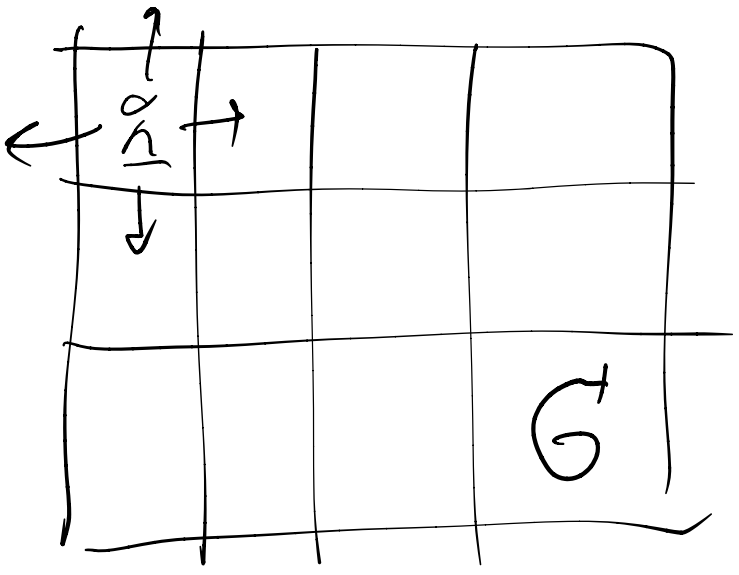
Agent



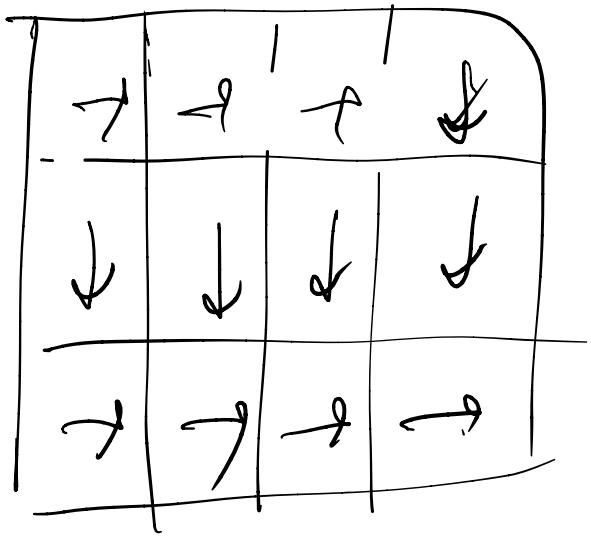
It follows a policy, π

$$\pi = P(A|S)$$

Example



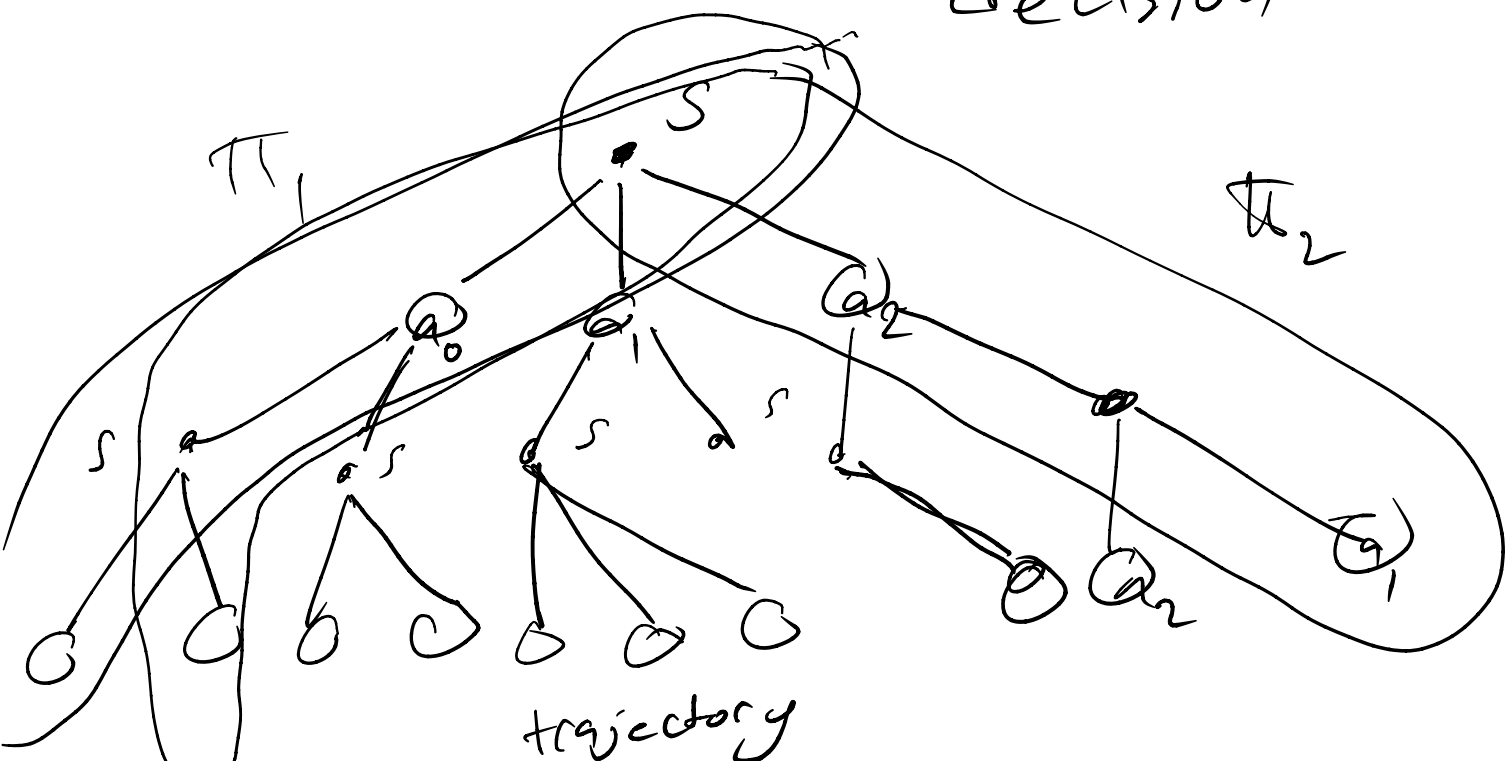
$\pi_1 =$



Objective of RL

We want to find "the best" policy.

Different policies take you to different path in a "decision tree".

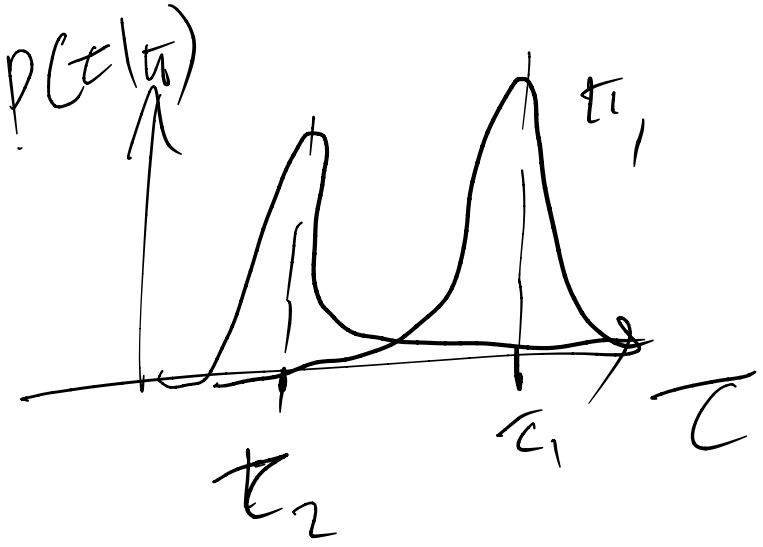


π_1 π_2

trajectory

$\tau = \{r_0, s_0, a_0, r_1, s_1, a_1, \ldots\}$

$p(\tau|\pi)$



$\gamma \in [0, 1]$

Let's define a Return G

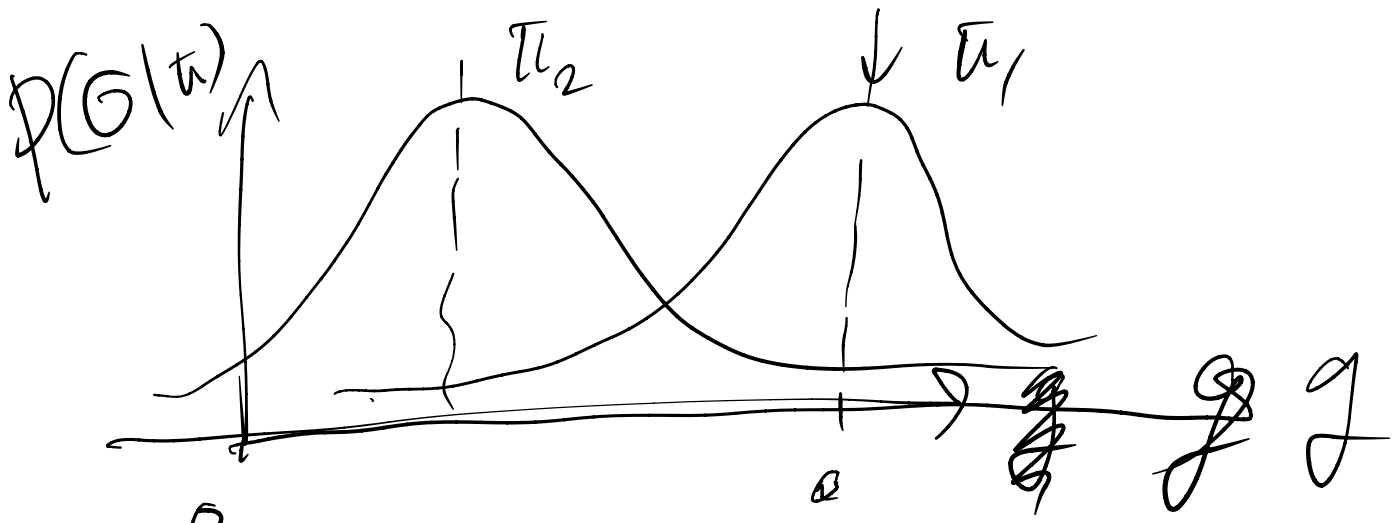
$$g = r_0 + \gamma r_1 + \gamma^2 r_2 + \dots + \gamma^T r_T$$

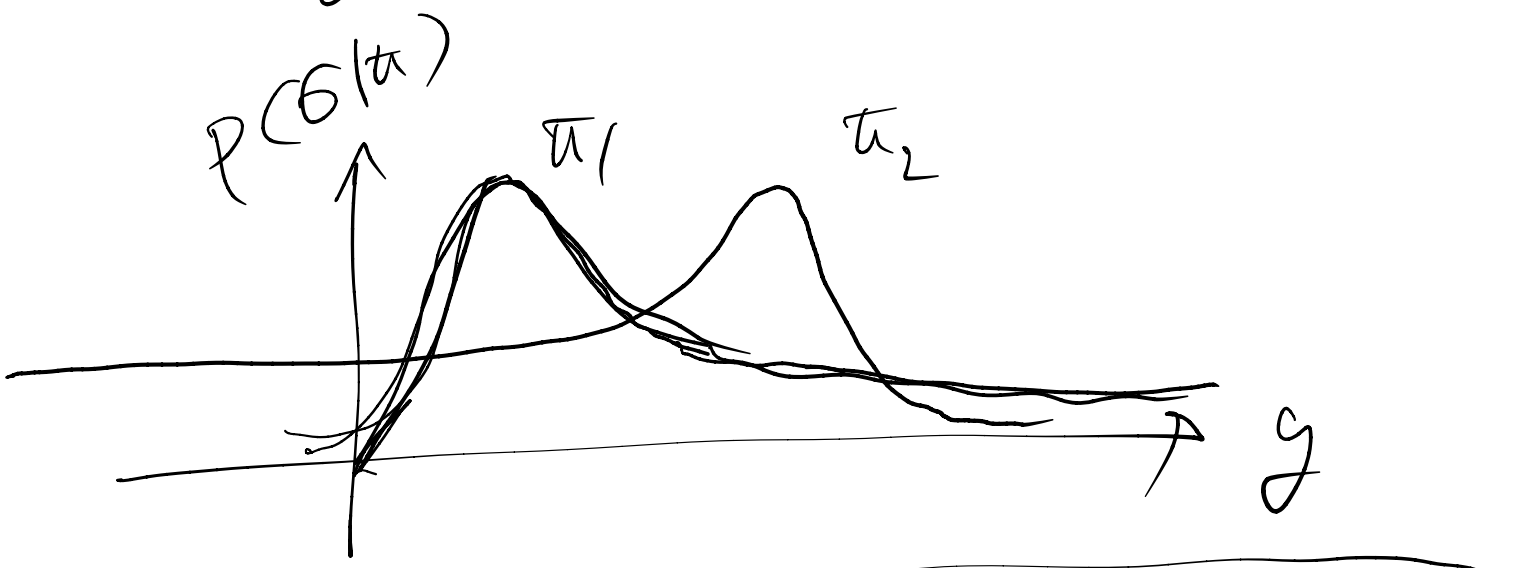
$$\tau_1 = \{ r_0, s_0, a_0, r_1, s_1, a_1, \dots, r_T \}$$

$$g_1 = r_0 + \gamma r_1 + \gamma^2 r_2 + \dots + \gamma^t r_t$$

* Trajectory τ_1 is better than τ_2, if g_1 is better than g_2

A policy π gives us a distribution of G; $P(G|\pi)$





In Classical RL,

We compare the mean

If $E[G|\pi_1] > E[G|\pi_2]$

then π_1 is better than π_2

How to find the best π?

π with with the highest $E[G|\pi]$ is the best π. π^*

How to find π^*

Let's define "Value"

$$V^{\pi}(s_t) = \mathbb{E}\left[G_t \mid \pi, S_t = s\right]$$

State-Value function

$$g = r_1 + \gamma r_2 + \gamma^2 r_3 + \dots \gamma^T r_T$$

$$g_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots + \gamma^{T-(t+1)} r_T$$

- Given a policy π

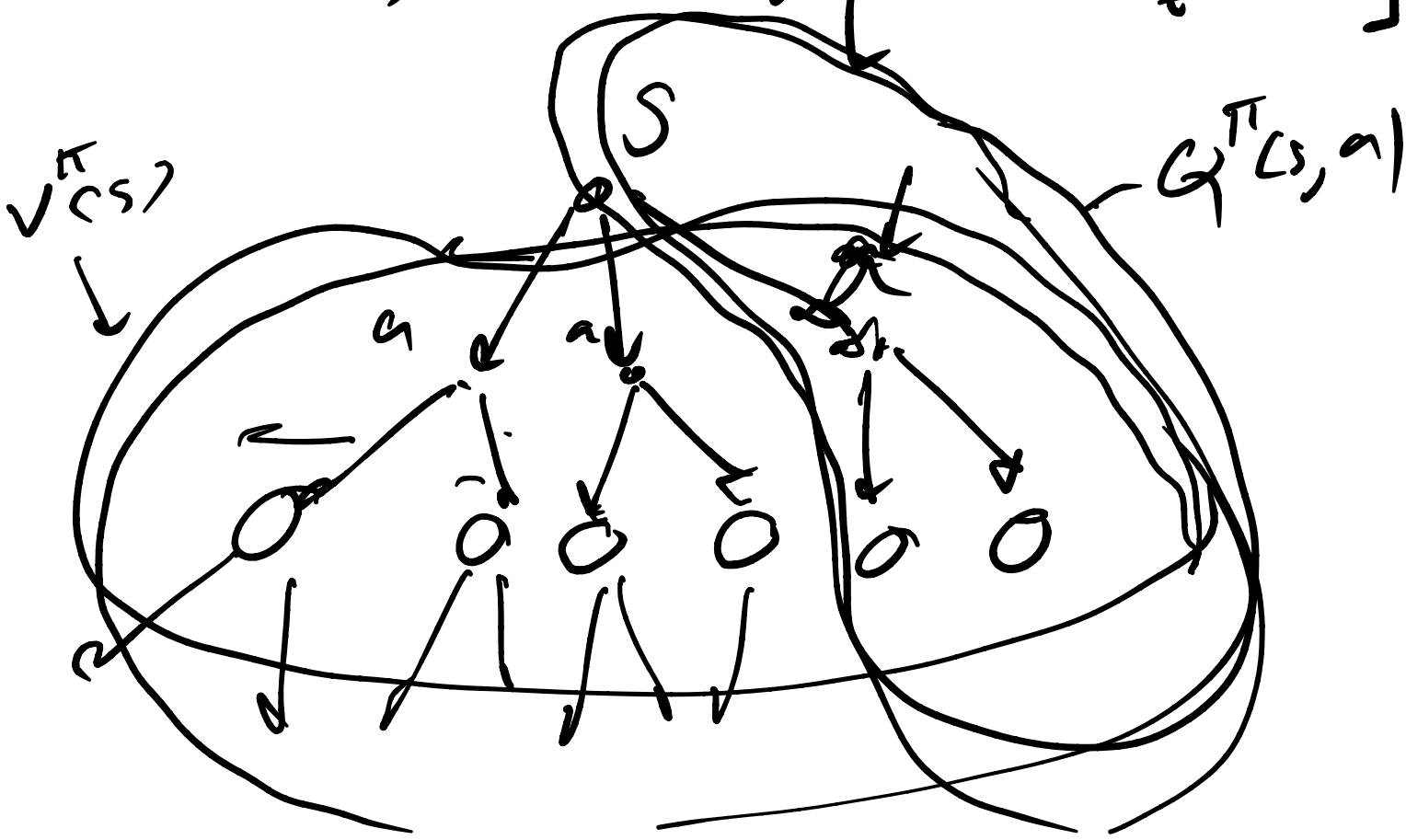
  if $V^{\pi}(s_1) > V^{\pi}(s_2)$

  We prefer s_1 to s_2

Similarly, we define

Q - Value

$$Q^{\pi}(s,a) = E_{\pi}\left[G_t \,\middle|\, \begin{matrix} s_t = s, \\ a_t = a \end{matrix} \right]$$



Optimal Policies & Optimal Value function

$\pi \geq \pi'$ (better)

if and only if

$$V^{\pi}(s) \geq V^{\pi'}(s)$$

$$\forall s \in S$$

There is always at least one policy that is <u>better than or equal to</u> all other policies. π^* (optimal policy)

π^* (optimal policy)

V^* (optimal value function)

$$V^*(s) = \max_\pi V^\pi(s) \quad \forall s \in S$$

$$Q^*(s,a) = \max_\pi Q^\pi(s,a) \quad \forall s, a \in S, A$$

$$V^\pi = E[G_t \mid S_t]$$

$$V^\pi = E[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots]$$

$$V^\pi = E[r_{t+1} + \gamma [r_{t+2} + \gamma r_{t+3} + \dots]]$$

$$V^{\pi}(s_t) = E\left[R_{t+1} + \gamma V^{\pi}(s_{t+1})\right]$$

$$V^{*}(s_t) = E^{\pi^{*}}\left[R_{t+1} + \gamma V^{*}(s_{t+1}) \,\middle|\, S_t = s\right]$$

$$Q^{*}(s_t, a_t) = E^{\pi^{*}}\left[R_{t+1} + \gamma V^{*}(s_{t+1}) \,\middle|\, S_t = s, A_t = a\right]$$

Bellman Equation for V^π

$$V^\pi(s) = E_{\pi,T}\left[G_t \mid S_t = s \right]$$

$$= E_{\pi,T}\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s \right]$$

$$= \sum_a \pi(a|s) \sum_{s'} \sum_r P(S_{t+1}=s', R_{t+1}=r \mid S_t = s) \cdot \left[r + \gamma E_\pi\left[G_{t+1} \mid S_{t+1} = s' \right] \right]$$

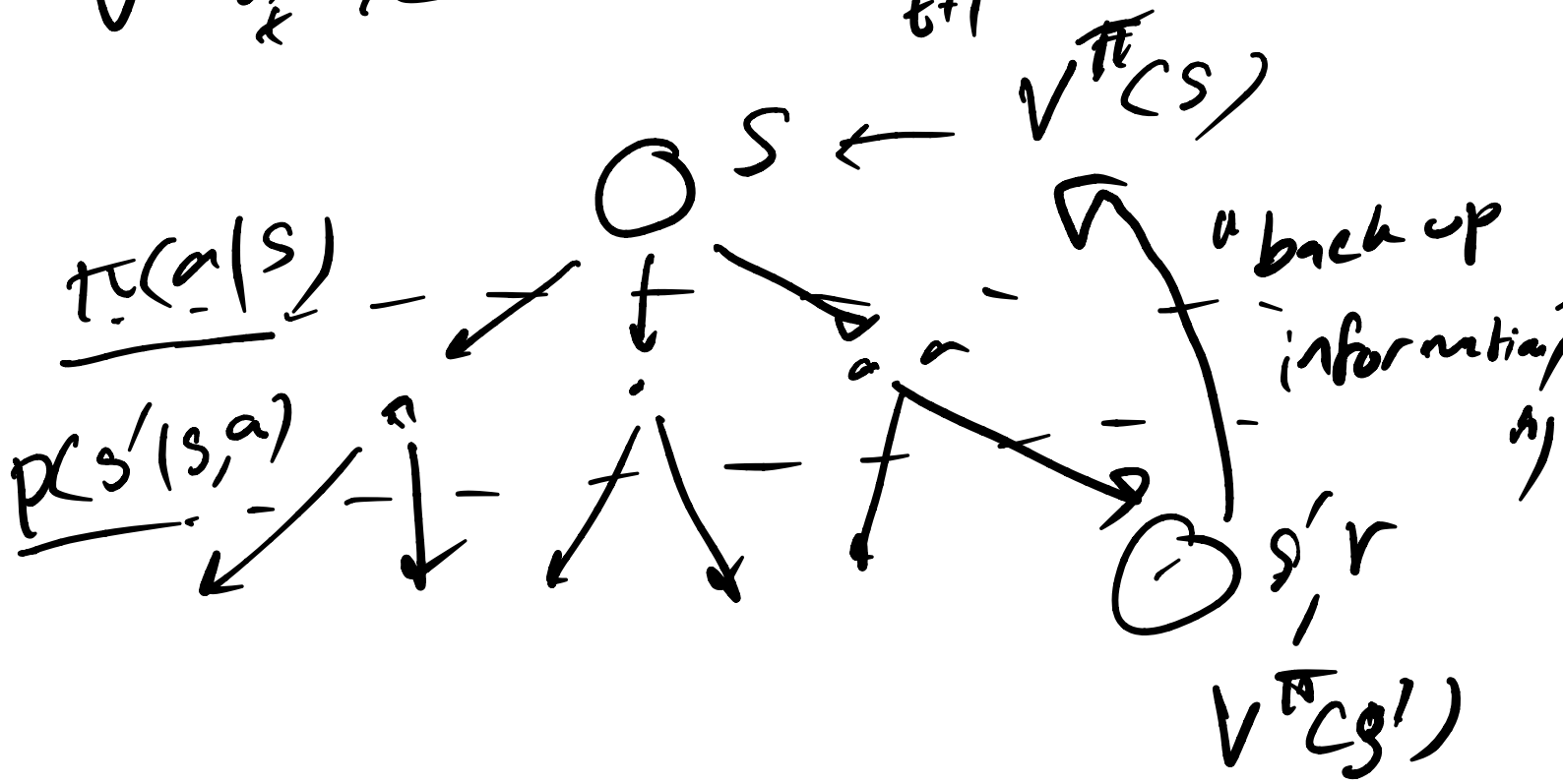
$$E_{P(X)}[X] = \sum_x P(X=x) \cdot x$$

$$V^\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\left[r + \gamma V^\pi(s') \right]$$

Bellman Eq →

$$V^\pi(s) = E_{\pi,T}\left[R + \gamma V^\pi(s') \right]$$

$$V^{\pi}(s_t) \approx r + \gamma V^{\pi}(s_{t+1})$$



Bellman Optimality Equation

$$V^*(s) = \max_a Q^*(s,a)$$

$$\boxed{V^*(s) = \max_a \sum_{s',r} p(s',r|s,a)\left[r + \gamma V^*(s')\right]}$$

$$Q^*(s,a) = \max_{a'} E_{\pi^*,\tau}\left[R + \gamma Q^*(s',a')\right]$$

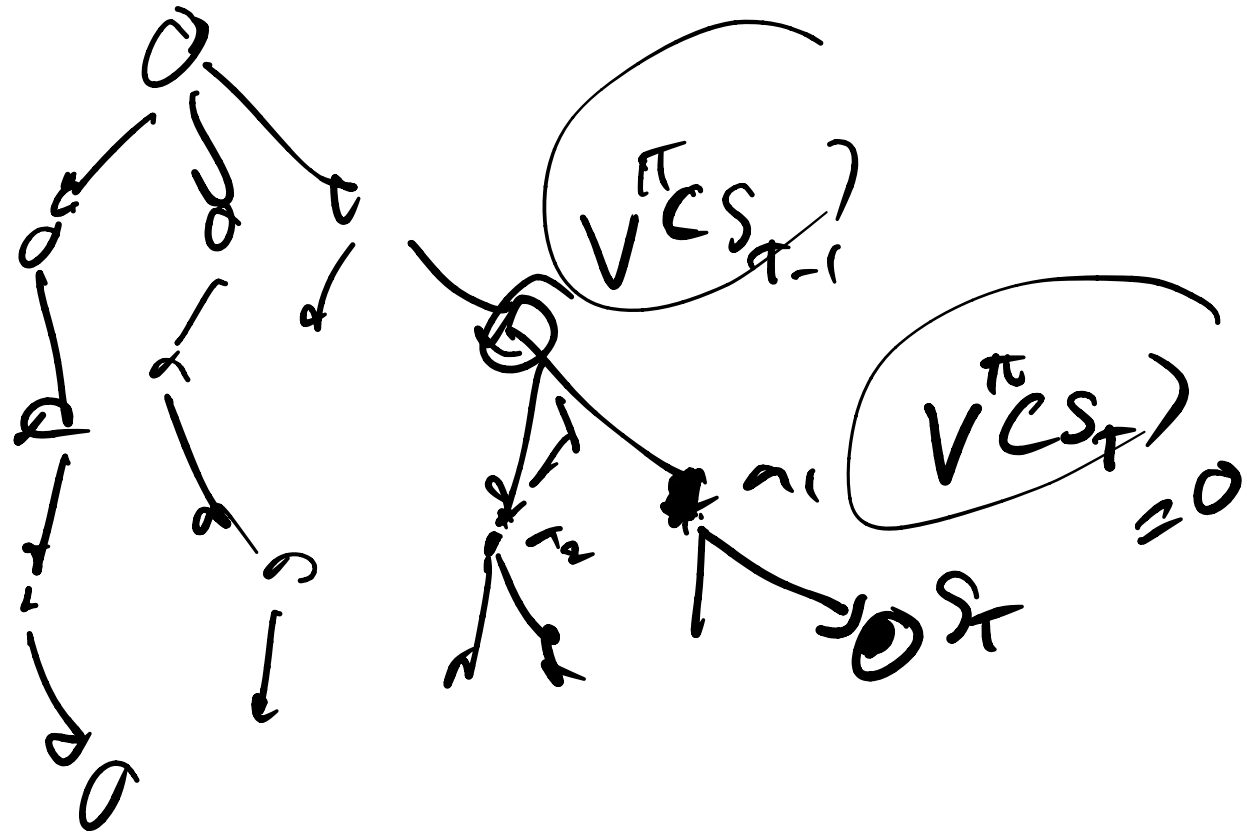
How to find $V^{\pi}(s)$, $V^{*}(s)$

$Q^{*}(s,a)$

for all s and a

DP = breaking problem into sub-problem

in a recursive manner.



Policy Evaluation ← Algorithm name

Find $V^{\pi}(s)$ $\quad Q^{\pi}(s,a)$

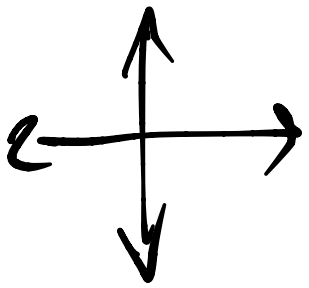
$$V^{\pi}(s) = \sum_{a} \pi(a|s) \sum_{s',r} p(s',r|s,a)\left[r + \gamma V^{\pi}(s')\right]$$

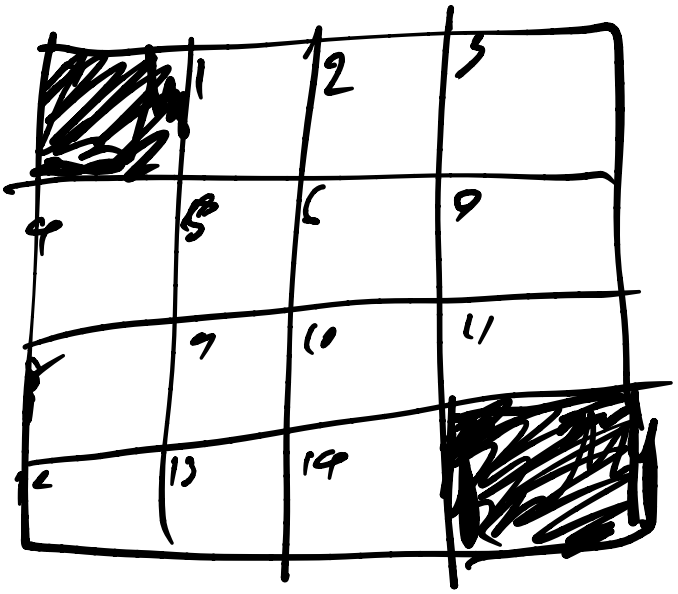
1.) Start with an initial $\hat{V}_0(s)$ $\forall s$

2.) Update $\hat{V}(s)$ with

$$\hat{V}_{k+1}(s) = \sum_{a} \pi(a|s) \sum_{s',r} p(s',r|s,a)\left[r + \gamma \hat{V}_k(s')\right]$$

3.) Repeat until Converge





$R_t = -1$

π_{random}

$V^{\pi_{random}}$

$V_{k=0}^{\pi}(s)$

$k=0$

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

$V = \sum \pi \sum P(s', r|s,a)[r + \gamma V_{k=1}^{\pi_{random}}]$

$k=1$

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

$K=2$

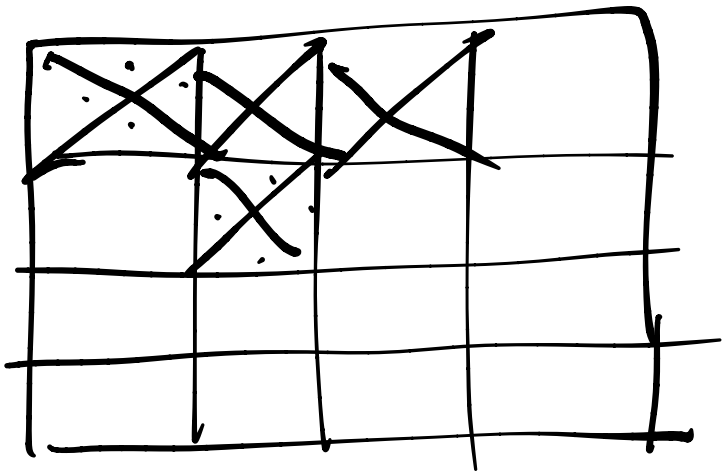
0.0	-1.7	-2.0	-2.0
-1.9	-2	-2	-2
-2	-2	-2	-1.7
-2	-2	-1.7	0.0

$$\hat{V}_{k=2}(S=1) = \frac{1}{4}\cdot\left[-1.0+\gamma \hat{V}_{k=1}^{-1}(S=2)\right]$$
$$+\frac{1}{4}\left[-1.0+\gamma \hat{V}_{k=1}(S=5)\right]$$
$$+\frac{1}{4}\left[-1.0+\gamma \hat{V}_{k=1}^{0}(S=T)\right]$$
$$+\frac{1}{4}\left[-1.0+\gamma \hat{V}_{k=1}^{-1}(S=1)\right]$$
$$= -\frac{1}{2}-\frac{1}{2}-\frac{1}{4}-\frac{1}{2}$$
$$= -1.75$$

$K = 2$

0.0	-16	-20	-22
-14	-18	-20	-20
-20	-20	-18	-14
-22	-20	-14	0.0

$\hat{Q}(s, a)$



Policy Improvement

theorem

if $Q^{\pi}(s, \pi'(s)) \geq V^{\pi}(s)$

then π' must be better than π

Policy Iteration

= Policy Evaluation + Policy Improvement

Value Iteration

= Policy Evaluation (only 1-step) + Policy Improvement