تمرین پنجم تئوری هوش مصنوعی

رادین شاه دائی ۴۰۱۱۰۶۰۹۶

سوال اول)

الف) این عبارت درست است. حل معادلات Bellman در روش value iteration، در واقع پیدا کردن مقدار fixed-point برای این معادلات است. با توجه به مطالب تدریس شده، مقدار بدست آمده توسط value iteration همان جواب Bellman equations است که V^* می‌باشد. در نتیجه اگر مقدار تابع policy به ازای یک T بزرگ به اندازه کافی تغییر نکند، می‌توان مطمئن بود که مقادیر V به V^* همگرا شده‌است و در نتیجه مقادیر تابع policy نیز همگرا شده است.

ب) این عبارت نادرست است. طبق مطالب گفته شده درون اسلایدها، به ازای انتخاب درست مقدار alpha، حتی به ازای sub-optimal policy، مقادیر Q نهایتا به مقادیر Q برای optimal policy همگرا خواهد شد. در نتیجه، می‌توان با استفاده policy نادرست در ابتدا، مقادیر Q را به ازای pi* را یاد گرفت.

ج) اگر learning rate = 1 باشد، این بدین معنی‌است که در هر مرحله از Q-learning، مقدار بدست آمده برای sample به جای Q قبلی قرار می‌گیرد. با توجه به اینکه MDP قطعی است، میانگین‌گیری از حرکات صورت‌گرفته خیلی معنا ندارد زیرا به ازای هر دو تایی s,a، می‌دانیم که یک حرکت رخ می‌دهد. در نتیجه، در صورت استفاده از alpha = 1 برای یک deterministic MDP، مقادیر Q به مقادیر بهینه همگرا خواهند شد.

د) پیچیدگی زمانی اجرای هر iteration از بخش iterative الگوریتم value iteration از $O(|S|^2|A|)$ می‌باشد. همچنین این پیچیدگی برای اجرای هر iteration از بخش iterative الگوریتم policy iteration، در بخش policy evaluation از $O(|S|^2)$ است. حال دقت کنید که جواب این سوال، بستگی به تعریف iteration دارد. اگر منظور طراح سوال از iteration در هر دو الگوریتم، مرحله آپدیت کردن V باشد، این گزاره صحیح خواهد بود. اما اگر منظور طراح سوال از iteration برای value iteration، مرحله آپدیت کردن V و برای policy iteration، مرحله آپدیت کردن pi باشد، این گزاره نادرست خواهد شد.

ه) نادرست است. فرض کنید در یک grid exploration هستیم که مقدار living reward به طور پیش‌فرض برابر 1- است. به طور شهودی، محرک تلاش می‌کند که کمترین زمان را در محیط چرخه بزند و زودتر به پاداش بزرگ برسد. اما اگر همه rewardها را با مقدار 1 جمع کنیم، چرخ زدن در محیط دیگر ضرری به محرک نخواهد زد و در نتیجه محرک از پذیرفتن ریسک گذر از خطر دوری می‌کند. دقت کنید که تحلیل صورت گرفته برای gridهایی می‌باشد که دارای terminal state می‌باشند.

در صورتی که بازی تا ابد ادامه داشته باشد، مقدار V برای تمامی stateها با مقدار ثابت $\sum_{i=0}^{\infty} c\gamma^i = \frac{c}{1-\gamma}$ جمع می‌شود. در این صورت، حرکتی که Q(s,a) را قبل از افزایش تمامی rewardها با مقدار c بیشینه می‌کرد، هنوز هم مقدار Q’(s,a) را بیشینه خواهد کرد. در نتیجه تفاوتی ایجاد نخواهد شد.

Value Iteration $\longrightarrow v_{K+1}(s) = \max\limits_{a} \sum\limits_{s'} T(s,a,s')\{R(s,a,s') + \gamma v_K(s')\}$ ۲) الف)

$$V_۰ = \begin{array}{ccc} ۰ & ۰ & ۰ \\ ۰ & ۰ & ۰ \end{array}$$

توجه کنید که حرکت مستقیم رو به دیوار نیز مجاز در نظر گرفته شده است.

در max ها، به ترتیب بالا، راست، پایین، چپ نوشته شده است.

first iter:

$v_۱\{۱,۱\} = \max\{۰,۰,۰,۰\} = ۰$

$v_۱\{۲,۱\} = \max\{۰,۰,۰,۰\} = ۰$

$$\begin{aligned} v_۱\{۱,۲\} = \max\{ & ۰٫۸(۰+۰) + ۰٫۱(-۵+۰) + ۰٫۱(۰+۰), \\ & ۰٫۸(-۵+۰) + ۰٫۱(۰+۰) + ۰٫۱(۰+۰), \\ & ۰٫۸(۰+۰) + ۰٫۱(۰+۰) + ۰٫۱(-۵+۰), \\ & ۰٫۸(۰+۰) + ۰٫۱(۰+۰) + ۰٫۱(۰+۰)\} = ۰ \leftarrow \end{aligned}$$

$$\begin{aligned} v_۱\{۲,۲\} = \max\{ & ۰٫۸(۰+۰) + ۰٫۱(۵+۰) + ۰٫۱(۰+۰), \\ & ۰٫۸(۵+۰) + ۰٫۱(۰+۰) + ۰٫۱(۰+۰), \\ & ۰٫۸(۰+۰) + ۰٫۱(۰+۰) + ۰٫۱(۵+۰), \\ & ۰٫۸(۰+۰) + ۰٫۱(۰+۰) + ۰٫۱(۰+۰)\} = ۴ \rightarrow \end{aligned}$$

$$\longrightarrow V_۱ = \begin{array}{ccc} ۰ & ۴ \rightarrow & ۰ \\ ۰ \leftarrow & ۰ & ۰ \end{array}$$

Second iter:

$v_۲\{۱,۱\} = \max\{۰,۰,۰,۰\} = ۰$

$v_۲\{۲,۱\} = ۰٫۸(۰+۴\gamma) + ۰٫۱(۰+۰) + ۰٫۱(۰+۰) = ۲٫۸۸ \rightarrow$ به وضوح حرکت به راست مطلوب است!

$$\begin{aligned} v_۲\{۱,۲\} = \max\{ & ۰٫۸(۰+۴\gamma) + ۰٫۱(-۵+۰) + ۰٫۱(۰+۰), \\ & ۰٫۸(-۵+۰) + ۰٫۱(۰+۴\gamma) + ۰٫۱(۰+۰), \\ & ۰٫۸(۰+۰) + ۰٫۱(۰+۰) + ۰٫۱(-۵+۰), \\ & ۰٫۸(۰+۰) + ۰٫۱(۰+۰) + ۰٫۱(۰+٤\gamma)\} = ۲٫۳۸ \uparrow \end{aligned}$$

$$\begin{aligned} v_۲\{۲,۲\} = \max\{ & ۰٫۸(۰+۴\gamma) + ۰٫۱(۵+۰) + ۰٫۱(۰+۰), \\ & ۰٫۸(۵+۰) + ۰٫۱(۰+٤\gamma) + ۰٫۱(۰+۰), \\ & ۰٫۸(۰+۰) + ۰٫۱(۰+۰) + ۰٫۱(۵+۰), \\ & ۰٫۸(۰+۰) + ۰٫۱(۰+۰) + ۰٫۱(۰+٤\gamma)\} = ۴٫۳۶ \rightarrow \end{aligned}$$

$$\longrightarrow v_۲ = \begin{array}{ccc} ۲٫۸۸ \rightarrow & ۴٫۳۶ \rightarrow & ۰ \\ & \uparrow & \\ ۰ & ۲٫۳۸ & ۰ \end{array}$$

subject: date:

third iter :

$v_۳[۱,۱] = ۰٫۸(۰+۲٫۸۸\gamma)+۰٫۱(۰+۲٫۳۸\gamma)+۰٫۱(۰+۰) = ۲٫۲۸۷۸ \ \textcircled{\uparrow}$ — وضع حرکت به بالا مطلوب است.

$v_۳[۲,۱] = ۰٫۸(۰+۴٫۳۶\gamma)+۰٫۱(۰+۲٫۸۸\gamma)+۰٫۱(۰+۰) = ۳٫۳۹۸۴ \ \textcircled{\rightarrow}$ — وضع حرکت رو به راست مطلوب است.

$$
\begin{aligned}
v_۳[۱,۲] = \max\{ & ۰٫۸(۰+۴٫۳۶\gamma)+۰٫۱(-۵+۰)+۰٫۱(۰+۰), \\
& ۰٫۸(-۵+۰)+۰٫۱(۰+۴٫۳۶\gamma)+۰٫۱(۰+۲٫۳۸\gamma) \\
& ۰٫۸(۰+۲٫۳۸\gamma)+۰٫۱(۰+۰)+۰٫۱(-۵+۰), \\
& ۰٫۸(۰+۰)+۰٫۱(۰+۴٫۳۶\gamma)+۰٫۱(۰+۲٫۳۸)\gamma\} = ۲٫۶۳۹۲ \ \textcircled{\uparrow}
\end{aligned}
$$

$v_۳[۲,۲] = ۰٫۸(۵+۰)+۰٫۱(۰+۴٫۳۶\gamma)+۰٫۱(۰+۲٫۳۸\gamma) = ۴٫۶۰۶۶ \ \textcircled{\rightarrow}$ — وضع حرکت رو به راست مطلوب است.

$$
\longrightarrow V_۳ = \begin{array}{ccccc} ۳٫۳۹۸۴ & \rightarrow & ۴٫۶۰۶۶ & \rightarrow & ۰ \\ \uparrow & & \uparrow & & \\ ۲٫۲۸۷۸ & & ۲٫۶۳۹۲ & & ۰ \end{array}
$$

ب) بهترین راهکار برای یادگیری π^* در این حالت استفاده از نسخه‌های مختلف Q-learning می‌باشد، به وضوح ساده‌ترین این روش‌ها، Passive Qlearning یا Off policy learning است.

ج) $\text{sample} = R(s,a,s')+\gamma v^{\pi}(s')$

$$v^{\pi}(s) \leftarrow v^{\pi}(s)+\alpha(\text{sample}-v^{\pi}(s))$$

حالت اول) مسیر اول $\longrightarrow (۱,۱)\rightarrow(۱,۲)\rightarrow(۲,۲)\rightarrow(۲,۳)$

$$
\begin{aligned}
V(۱,۱) &\leftarrow ۰+\alpha(۰+\gamma\times۰-۰)=۰ \\
V(۱,۲) &\leftarrow ۰+\alpha(۰+\gamma\times۰-۰)=۰ \\
V(۲,۲) &\leftarrow ۰+\alpha(۵+۰٫۹\times۰-۰)=۰٫۵
\end{aligned}
$$

مسیر دوم $\longrightarrow (۱,۱)\rightarrow(۱,۲)\rightarrow(۱,۳)$

$$
\begin{aligned}
V(۱,۱) &\leftarrow ۰+\alpha(۰+\gamma\times۰-۰)=۰ \\
V(۱,۲) &\leftarrow ۰+\alpha(-۵+\gamma\times۰-۰)=-۰٫۵
\end{aligned}
$$

احتمالاً منظور طراح این ترتیب مسیرها نبوده است :)

حالت دوم :

مسیر اول —— $(۱,۱) \rightarrow (۱,۲) \rightarrow (۱,۳)$

$$V(۱,۱) \leftarrow ۰ + \alpha(۰ + \gamma \times ۰ - ۰) = ۰$$

$$V(۱,۲) \leftarrow ۰ + \alpha(-۵ + \gamma \times ۰ - ۰) = -۰٫۵$$

مسیر دوم —— $(۱,۱) \rightarrow (۱,۲) \rightarrow (۲,۲) \rightarrow (۲,۳)$

$$V(۱,۱) \leftarrow ۰ + \alpha(۰ + \gamma \times -۰٫۵ - ۰) = -۰٫۰۴۵$$

$$V(۱,۲) \leftarrow -۰٫۵ + \alpha(۰ + \gamma \times ۰ + ۰٫۵) = -۰٫۴۵$$

$$V(۲,۲) \leftarrow ۰ + \alpha(۵ + \gamma \times ۰ - ۰) = ۰٫۵$$

$$V^{\pi} : \begin{matrix} ۰ & ۰٫۵ & ۰ \\ -۰٫۰۴۵ & -۰٫۴۵ & ۰ \end{matrix}$$

الف)

۳) معادلات بلمن بصورت زیر هستند:

$$۱)\ \forall s : V^*(s) = \max_a Q^*(s,a)$$

$$۲)\ \forall s,a : Q^*(s,a) = \sum_{s'} T(s,a,s')\{R(s,a,s') + \gamma V^*(s')\}$$

می‌توان این دو معادله را بصورت زیر بازنویسی کرد:

$$\forall s,a : Q^*(s,a) = \sum_{s'} T(s,a,s')\{R(s,a,s') + \gamma \max_{a'} Q^*(s',a')\}$$

با حل این معادلات، ماتریس Q تشکیل شده (با اندازه $|S| \times |A|$) و به تمامی $Q(s,a)$ ها دسترسی خواهیم داشت.

برای حل این معادلات، می‌توان از روش iterative استفاده کرد. دقت کنید که اگر معادلات را به صورت ماتریسی بنویسیم، ماتریس Q^*، نقطه ثابت این معادله خواهد شد.

$$\text{Iteration} : Q_{K+1}(s,a) \leftarrow \sum_{s'} T(s,a,s')\{R(s,a,s') + \gamma \max_{a'} Q_K(s',a')\}$$

در روش‌های RL، به توابع T و R دسترسی نداریم، در نتیجه این مرحله iteration را با سمپل‌گیری باید انجام دهیم.

مشاهده

از سمپل‌های قبلی داریم.

✓ به منظور این کار، از exp moving avg مجددا استفاده می‌کنیم:

$$sample = R(s,a,s') + \gamma \max_{a'} Q(s',a')$$

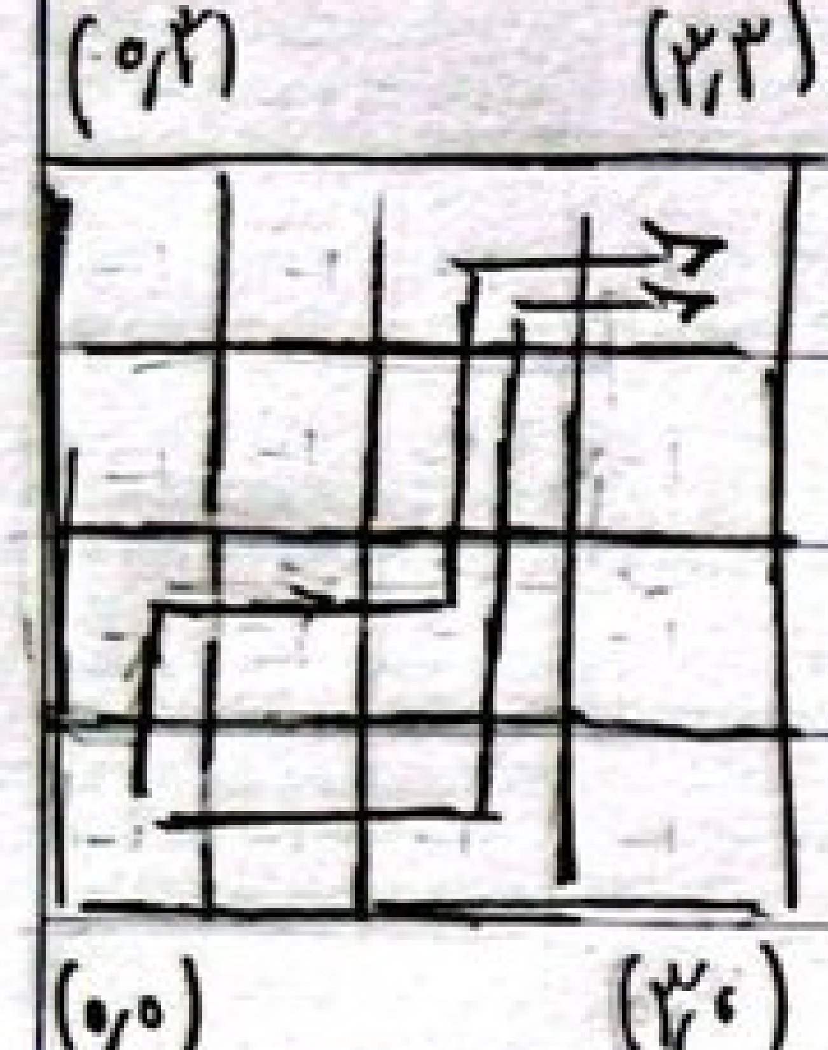
$$Q(s,a) \leftarrow Q(s,a) + \alpha\left(sample - Q(s,a)\right)$$

— در یک صفحه ۴×۳ نیز می‌توان از روش‌های Active یا Passive برای یادگیری مقادیر Q استفاده کرد.

— در روش Passive یک fixed policy داریم که لزوما optimal نیست، حرکت‌ها طبق آن انجام می‌شوند.

— در روش‌های Active تابع π نیز تغییر می‌کند و در واقع exploration صورت می‌گیرد.

ب) فرض کنید در ۲ iteration، از مسیرهای زیر حرکت کردیم:



ابتدا مسیر بالایی را طی کردیم. طبق تعریف temp diff که در بخش الف داشتیم، مقادیر Q به صورت زیر می‌شود:

$Q((۰,۰),\uparrow) = -۰٫۱$

$Q((۰,۱),\rightarrow) = -۰٫۱$

$Q((۱,۱),\rightarrow) = -۰٫۱$

$Q((۲,۱),\uparrow) = -۰٫۱$

$Q((۲,۲),\uparrow) = -۰٫۱$

$Q((۲,۳),\rightarrow) = ۱$

سپس مسیر پایینی را طی می‌کنیم:

$Q((۰,۰),\rightarrow) = -۰٫۱$

$Q((۱,۰),\rightarrow) = -۰٫۱$

$Q((۲,۰),\uparrow) = ۰ + ۰٫۱(-۱ + ۰٫۹ \max Q((۲,۱)) - ۰) = -۰٫۱$

$Q((۲,۱),\uparrow) = -۰٫۱ + ۰٫۱(-۱ + ۰٫۹ \max Q(۲,۲) + ۰٫۱) = -۰٫۱۹$

$Q((۲,۲),\uparrow) = -۰٫۱ + ۰٫۱(-۱ + ۰٫۹ \max Q(۲,۳) + ۰٫۱) = -۰٫۱$

$Q((۲,۳),\rightarrow) = ۱ + ۰٫۱(۱۰ + ۰٫۹ \max Q(۳,۳) - ۱) = ۱٫۹$

ج) در روش ε-greedy، عملی با احتمال ε به صورت رندوم عمل می‌شد. این‌طور می‌توانیم محیط را explore کنیم و بفهمیم π مورد استفاده تا چه حد درست است.

ε را مشابه simulated annealing به مرور زمان می‌توان کم کرد.

with prob ε act randomly
with prob 1-ε act on current policy
$\arg\max \hat{Q}(s,a)$

در واقع در این روش، داریم: → ۱) explore
۲) exploit

۴) الف)

$a' = \arg\max_{a} Q^*(s,a) \longrightarrow v^*(s) = Q^*(s,a')$

$\pi(s) = \arg\max_{a} \tilde{Q}(s,a) \longrightarrow \tilde{Q}(s,\pi(s)) \geq \tilde{Q}(s,a')$ ③

$\max_{s,a} |Q^*(s,a) - \tilde{Q}(s,a)| \leq \varepsilon \longrightarrow Q^*(s,a') - \tilde{Q}(s,a') \leq \varepsilon$ ①

$\longrightarrow \tilde{Q}(s,\pi(s)) - Q^*(s,\pi(s)) \leq \varepsilon$ ②

① $\longrightarrow v^*(s) - \tilde{Q}(s,a') \leq \varepsilon$

③ $\longrightarrow v^*(s) - \tilde{Q}(s,\pi(s)) \leq \varepsilon$

② $\longrightarrow v^*(s) - Q^*(s,\pi(s)) \leq 2\varepsilon$ ■

ب)

$v^*(s) - v_\pi(s) = f(s)$

$\hookrightarrow v^*(s) - Q^*(s,\pi(s) + Q^*(s,\pi(s)) - v_\pi(s) \leq 2\varepsilon + Q^*(s,\pi(s)) - v_\pi(s)$

$Q^*(s,\pi(s)) = \sum_{s'} T(s,\pi(s),s')\{R(s,\pi(s),s') + \gamma v^*(s')\} = R(s,a) + \gamma \sum T(s,\pi(s),s') v^*(s')$

$v_\pi(s) = \sum_{s'} T(s,\pi(s),s')\{R(s,\pi(s),s') + \gamma v_\pi(s')\} = R(s,a) + \gamma \sum T(s,\pi(s),s') v_\pi(s')$

$\longrightarrow Q^*(s,\pi(s)) - v_\pi(s) = \gamma \sum T(s,\pi(s),s')\{v^*(s') - v_\pi(s)\} = \gamma E\{v^*(s') - v_\pi(s')\}$

$E\{v^*(s') - v_\pi(s')\} \leq \max_{s'}\{v^*(s') - v_\pi(s)\} = v^*(s'') - v_\pi(s'') = f(s'')$

$\longrightarrow \forall s \exists s''$ such that : $f(s) \leq 2\varepsilon + \gamma f(s'')$

$\longrightarrow f(s) \leq 2\varepsilon \sum_{i=0}^{n} \gamma^i + \underbrace{\gamma^{n+1} f(s_f)} \longrightarrow$ این جمله صفر می‌شود چون $\gamma < 1$ و همچنین $v^*(s)$ و $v_\pi(s)$ کران‌دار هستند.

$\lim_{n \to \infty} : v^*(s) - v_\pi(s) \leq \frac{2\varepsilon}{1-\varepsilon}$ ■

$v^*(s_1) = \max\{Q^*(s_1, stay), Q^*(s_1, go)\}$ (ج

$v^*(s_۲) = ۲\varepsilon + \gamma v^*(s_۲) \rightarrow v^*(s_۲) = \frac{۲\varepsilon}{1-\gamma}$

$Q^*(s_1, stay) = 0 + \gamma v^*(s_1)$

$Q^*(s_1, go) = ۲\varepsilon + \gamma v^*(s_۲) = \frac{۲\varepsilon}{1-\gamma}$

① $v^*(s_1) = Q^*(s_1, stay) \xrightarrow[\text{or } v^*(s_1)=0]{} \gamma = 1$ ✗ $\rightarrow v^*(s_1) = Q^*(s_1, go) = \frac{۲\varepsilon}{1-\gamma}$

$\rightarrow Q^*(s_1, stay) = \frac{۲\varepsilon\gamma}{1-\gamma}$

$\|Q^* - \tilde{Q}\| \leq \varepsilon \xrightarrow{ش} \tilde{Q}(s_1, stay) = Q^*(s_1, stay) + \varepsilon = \varepsilon\left(\frac{۲\gamma}{1-\gamma} + 1\right) = \varepsilon \times \frac{1+\gamma}{1-\gamma}$ ① (۲

$\rightarrow \tilde{Q}(s_1, go) = Q^*(s_1, go) - \varepsilon = \varepsilon\left(\frac{۲}{1-\gamma} - 1\right) = \varepsilon \times \frac{1+\gamma}{1-\gamma}$ (۲)

(۲),① $\rightarrow \tilde{Q}(s_1, stay) = \tilde{Q}(s_1, go)$

$\hookrightarrow \pi(s_1) = stay$

$v_\pi(s_1) = 0 + \gamma v_\pi(s_1) \rightarrow \underline{v_\pi(s_1) = 0}$

$\rightarrow v_\pi(s_1) = v^*(s_1) - \frac{۲\varepsilon}{1-\gamma}$ ■

ه) الف)

total reward : $Q^*(s_0,a_1) = 0 + \gamma v^*(s_1)$

$v^*(s_1) = 1 + \gamma v^*(s_1) \rightarrow v^*(s_1) = {}^1/_{1-\gamma}$

$\Big\} \; Q^*(s_0,a_1) = {}^{\gamma}/_{1-\gamma}$

ب)

total reward : $Q^*(s_0,a_2) = {}^{\gamma^2}/_{1-\gamma} + \gamma v^*(s_2) \longrightarrow Q^*(s_0,a_2) = {}^{\gamma^2}/_{1-\gamma}$

$v^*(s_2) = 0$

با توجه به اینکه $0 < \gamma < 1$ ، داریم $\gamma^2 < \gamma$ ، پس $Q^*(s_0,a_1) > Q^*(s_0,a_2)$ ← عمل بهینه a_1 است.

ج)

$v_0(s_i) = 0 \;, \; \forall i$

$v_1(s_0) = \max\left(0 + \gamma v_0(s_1), \frac{\gamma^2}{1-\gamma} + \gamma v_0(s_2)\right) = \frac{\gamma^2}{1-\gamma} \longrightarrow$ a_2 را انتخاب می‌کند

$v_1(s_1) = 1$

$v_1(s_2) = 0$

$\longrightarrow v_n(s_1) = 1 + \gamma + \gamma^2 + \dots + \gamma^n = \frac{1-\gamma^{n+1}}{1-\gamma}$

$v_n(s_2) = 0$

$\Big\} \; v_n(s_0) = \max\left\{\gamma v_{n-1}(s_1), \frac{\gamma^2}{1-\gamma}\right\}$

$\longrightarrow v_n(s_0)$ finds optimal action when $\longrightarrow \gamma v_{n-1}(s_1) \geq \frac{\gamma^2}{1-\gamma}$

$$\longrightarrow 1 - \gamma^n \geq \gamma$$

$$1 - \gamma \geq \gamma^n$$

$$\log(1-\gamma) \geq n \log(\gamma)$$

$$\longrightarrow n^* \geq \frac{\log(1-\gamma)}{\log \gamma} \;\blacksquare$$