



دانشگاه تهران  
پردیس دانشکده‌های فنی  
دانشکده برق و کامپیوتر



**گزارش تمرین شماره 3**

**درس یادگیری تعاملی**

**پاییز 1401**

امیرحسین بیرژندی

...

810198367

...

## فهرست

چکیده.....	3
سوال 1 - سوال تئوری.....	4
سوال 2 - سوال دستی.....	7
هدف سوال.....	7
سوال 1.....	8
سوال 2.....	11
سوال 3.....	14
سوال 4.....	16
سوال 3 - سوال پیاده سازی.....	17
سوال 1.....	17
سوال 2.....	21
سوال 3.....	26
منابع.....	28

## چکیده

---

هدف از این تمرین آشنایی با مسائل MDP، مدل‌سازی و حل آنهاست. در این تمرین دو الگوریتم مهم value iteration و policy iteration را با شرایط مختلفی پیاده‌سازی و تحلیل می‌کنیم.

## سوال 1 - سوال تئوری

در این فیتنس اپ باتوجه به استیت فرد برنامه‌ها تنظیم می‌شوند.

استیت فرد شامل موارد زیر می‌باشد:

**میزان اضافه وزن با توجه به شاخص BMI:** این فیتنس اپ با دریافت وزن و قد فرد BMI را محاسبه می‌کند و مقدار محاسبه شده را برای نوع برنامه و رژیم در نظر می‌گیرد. دلیل وجود این شاخص نیز مشخص است فردی که اضافه وزن دارد باید ملاحظات در درجه سختی تمرین داشته باشد و برای مثال حرکاتی که به زانو و یا کمر فشار می‌آورد را کمتر پیشنهاد دهد.

**علاقه شخصی به نوع تمرین:** این فیتنس اپ در ابتدای هر دوره تمرینی از فرد علاقه شخصی وی را در مورد نوع تمرین سوال می‌کند که بین تمارین هوازی و بی هوازی به کدام یک علاقه بیشتری داری؟ و پس از دریافت پاسخ کمی تمارین خود را نسبت به علاقه شخص بایاس می‌کند تا بتواند رضایت فرد را جلب کند. اما این مورد نیز با توجه به فیدبک های دریافتی در حین یک دوره تمرین تغییر می‌کند.

**وجود مصدومیت در ناحیه خاص:** این فیتنس اپ قبل از هر تمرین از فرد راجع به وضعیت جسمانی یا همان مصدومیت پرسش می‌کند و اگر مصدومیتی وجود داشته باشد با توجه به درجه آن نوع برنامه را تنظیم می‌کند که در صورت لزوم ممکن است در آن روز تمرینی را پیشنهاد ندهد.

**توانایی فرد تمرین کننده:** یکی از مواردی که این فیتنس اپ در نظر می‌گیرد میزان توانایی فرد است که باتوجه به میزان رضایت فرد از تمارین و سوخت و ساز وی تعیین می‌شود.

**تمارین انجام شده تا کنون:** برای این فیتنس اپ بسیار مهم است که تمارین را پخش کند و سعی کند در هر روز بر روی قسمت های خاصی از بدن تمرکز کند. در نتیجه از با دانستن تمرین های انجام شده در روز های قبلی برنامه امروز را تعیین می‌کند.

**حساسیت و یا عدم علاقه به یک غذای خاص:** اگر فرد به یک غذای خاص حساسیت داشته باشد و یا علاقه‌ای نداشته باشد از آن غذا پرهیز می‌شود.

اکشن های فیتنس اپ به صورت زیر هستند.

**دادن تمرین:** اصلی ترین اکشن این اپلیکیشن دادن یک برنامه مناسب است. که موارد زیر در آن در نظر گرفته می شود.

1- مدت زمان تمرین: با توجه به توانایی فرد و نوع تمرین تعیین می شود.

2- درجه سختی تمرین: با توجه به توانایی، فرد علاقه شخصی و شاخص BMI تعیین می شود.

3- نوع تمرین (هوازی یا بی هوازی یا در صورت مصدومیت عدم تمرین)

4- تمرکز عضله: با توجه به تمارین انجام شده تا کنون تعیین میشود.

**دادن رژیم غذایی:** این اپلیکیشن یک رژیم غذایی توصیه می کند که در ابتدا با توجه به استیت حساسیت یا عدم علاقه به غذا فرد تعیین می شود. اما در طول دوره با دریافت فیدبک از فرد نسبت به میزان علاقه و حجم تغییر می کند. در این رژیم دو مورد در نظر گرفته می شود.

1- نوع غذا: مقدار چربی، مقدار کالری و مقدار پروتئین در آن موثر هستند.

2- حجم غذا

پاداش ها در این فیتنس اپ شامل موارد زیر هستند.

**میزان سوخت و ساز:** با وجود میزان سوخت و ساز دریافتی از فرد (با فرض وجود ساعت ورزشی) در پاداش دریافتی به موثر بودن برنامه تمرینی و رژیم غذایی پی ببریم. برای مثال اگر فردی سوخت و ساز کمی دارد نشان دهنده این است که سطح برنامه برای وی ساده است و باید مجازات بشویم تا برنامه سختی را پیشنهاد دهیم.

**علاقه فرد از تمرین انجام شده و غذای پیشنهاد داده شده:** با دریافت فیدبک از فرد نسبت به تمرین انجام شده و با مرور زمان و پیدا کردن وجه های مشترک تمارین می توانیم به یک جمع بندی راجع به علاقه فرد درباره مدل تمرین داشته باشیم. مشخصا رضایت مندی فرد موجب دریافت پاداش و گله فرد موجب دریافت مجازات می شود. دقیقا همه موارد برای غذا نیز مشابه است. گله فرد ممکن است از به وجود آمدن مصدومیت باشد که این شامل یک مجازات بسیار بالا می شود.

**انتقال بین استیت ها** ابتدا اصلا مشخص نیست زیرا علاوه بر موارد تشکیل دهنده استیت موارد دیگری وجود دارد که برنامه فقط با گذشت زمان از آن ها خبردار می شود. برای مثال فردی که ژن قوی ای دارد ممکن است با وجود اضافه وزن فراوان با کیفیت بسیار خوبی تمرین کند و یا فردی ممکن است شرایط

زندگی وی به نوع تمرین کردن او اثر بگذارد. اما با گذشت زمان احتمال انتقال بین استیت ها تقریبا به یک روال استاندارد تغییر می کند.

## سوال 2 - سوال دستی

### هدف سوال

در این بخش هدف سوال آشنایی با سیاست‌های value iteration و policy iteration است. در policy iteration ابتدا یک سیاست را ارزیابی می‌کنیم و سپس آن را ارتقا می‌دهیم و دوباره این روند را تکرار می‌کنیم. Psuedo code آن در تصویر زیر قابل رویت می‌باشد.

#### 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:

$\Delta \leftarrow 0$

Loop for each  $s \in \mathcal{S}$ :

$v \leftarrow V(s)$

$V(s) \leftarrow \sum_{s',r} p(s', r | s, \pi(s)) [r + \gamma V(s')]$

$\Delta \leftarrow \max(\Delta, |v - V(s)|)$

until  $\Delta < \theta$  (a small positive number determining the accuracy of estimation)

##### 3. Policy Improvement

*policy-stable*  $\leftarrow$  true

For each  $s \in \mathcal{S}$ :

*old-action*  $\leftarrow \pi(s)$

$\pi(s) \leftarrow \operatorname{argmax}_a \sum_{s',r} p(s', r | s, a) [r + \gamma V(s')]$

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

## سوال 1

در این بخش خواسته شده با  $\text{discount factor} = 0$  به وسیله policy iteration مسئله را حل کنیم.  
حال مقادیر ارزش ها را برابر 0 و یک سیاست رندوم در نظر می گیریم.

$$V(s) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \pi_0 = \begin{matrix} & u & d & l & r \\ \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix} & \end{matrix}$$

ابتدا این سیاست را ارزیابی می کنیم. برای این کار از فرمول رابطه زیر استفاده می شود.

$$V(s) = \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V(s')]$$

از آنجایی که discount factor برابر صفر است فرمول بالا به صورت زیر تبدیل می شود.


$$V(s) = \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a R_{ss'}^a$$

برای ارتقا سیاست نیز باید از رابطه زیر استفاده کنیم.

$$\operatorname{argmax}_a \sum_{s'} P_{ss'}^a R_{ss'}^a$$



در تصویر زیر یک ایتريشن از یک ارزیابی سیاست را مشاهده می کنید. این کار را تا جایی ادامه می دهیم که مقادیر ارزش ها تقریباً همگرا شوند.



$$V(s) = \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V(s')]$$

$$V(s_0) = \frac{1}{4} \times 1 \times 0 + \dots = 0$$

$$V(s_1) = \frac{1}{4} \times 1 \times [-10] = -2.5$$

$$V(s_2) = \frac{1}{4} \times 1 \times [-10] = -2.5$$

$$V(s_3) = \frac{1}{4} \times 1 \times 0 = 0$$

$$V(s_4) = \frac{1}{4} \times 1 \times [-10] = -2.5$$


$$V(s_5) = \frac{1}{4} \times 1 \times 10 = +2.5$$

$$V(s_6) = 0$$

$$V(s_7) = 0$$

$$V(s_8) = \frac{1}{4} \times 1 \times 10 = +2.5$$

$$V = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow V = \begin{bmatrix} 0 \\ -2.5 \\ -2.5 \\ 0 \\ -2.5 \\ +2.5 \\ 0 \\ 0 \\ +2.5 \end{bmatrix}$$



$$V(s) = \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a R_{ss'}^a$$

$$V(s_0) = 0$$

$$V(s_1) = \frac{1}{4} \times 1 \times [-10] = -2.5$$

$$V(s_2) = \frac{1}{4} \times 1 \times [-10] = -2.5$$

$$V(s_3) = \frac{1}{4} \times 1 \times [0] = 0$$

$$V(s_4) = \frac{1}{4} \times 1 \times [-10] = -2.5$$

$$V(s_5) = \frac{1}{4} \times 1 \times [10] = +2.5$$

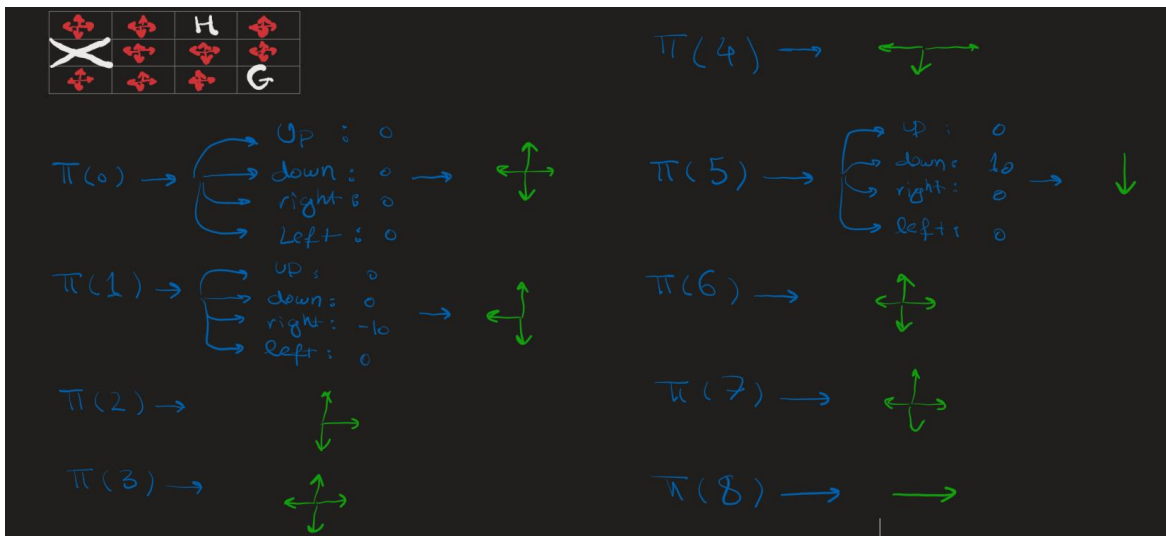
$$V(s_6) = 0$$

$$V(s_7) = 0$$

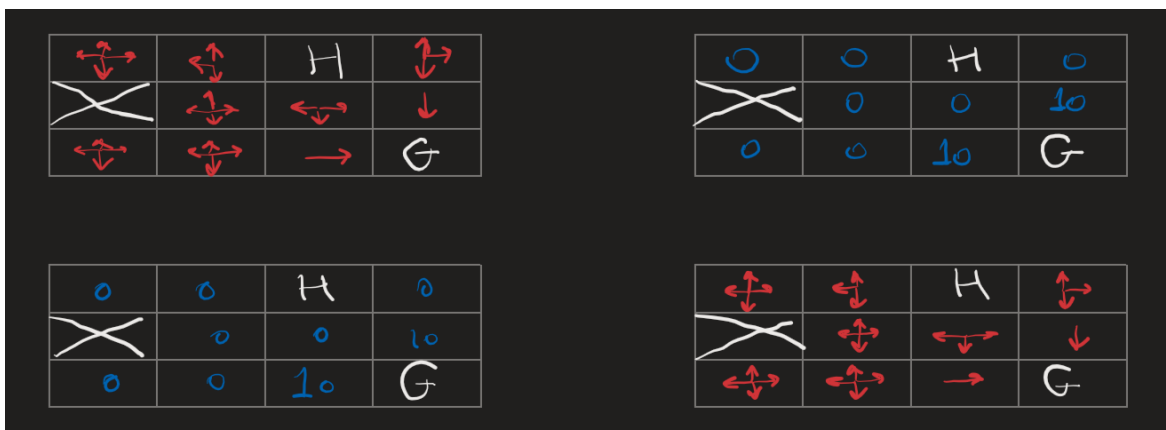
$$V(s_8) = \frac{1}{4} \times 1 \times [10] = +2.5$$

$$V(s) = \begin{bmatrix} 0 \\ -2.5 \\ -2.5 \\ 0 \\ -2.5 \\ +2.5 \\ 0 \\ 0 \\ +2.5 \end{bmatrix} \Rightarrow V = \begin{bmatrix} 0 \\ -2.5 \\ -2.5 \\ 0 \\ -2.5 \\ +2.5 \\ 0 \\ 0 \\ +2.5 \end{bmatrix}$$

حال با توجه به اینکه ارزش استیت ها تغییری نکرده است اولین مرحله از ارزیابی سیاست ما به اتمام رسیده است و به سراغ بهبود سیاست می رویم.



بهبود سیاست ما نیز برای سری اول به اتمام رسید. حال این سیاست را ارزیابی می‌کنیم. محاسبات جزئی دیگر آورده نمی‌شود و صرفاً پاسخ‌های نهایی نمایش داده می‌شود. از آنجایی که discount factor برابر صفر است. مرحله ارزیابی سیاست 1 بار بیشتر انجام نمی‌شود. در نهایت سیاست نهایی در شکل زیر نمایش داده شده است.



با توجه به نتایج بالا مشاهده می‌کنیم با وجود discount factor برابر با صفر، نمی‌توانیم از ارزش استیت‌های کناری خود خبردار شویم و فقط استیت‌هایی که در کنار ترمینال استیت‌ها که با ورود به آن‌ها پاداش دریافت می‌کنند دارای ارزش می‌شوند. در استیت‌های کنار hell نیز سیاست به گونه‌ای تعیین شده است که امکان ورود به hell را نداشته باشیم. پس با discount factor صفر قبل از 5 ایتريشن به سیاست نهایی همگرا می‌شویم.

## سوال 2

در این بخش با discount factor برابر 0.9 مسئله را به کمک policy iteration حل می کنیم. همانند قبل مقادیر ارزش ها را برابر صفر و یک سیاست رندوم انتخاب می کنیم.

$$V(s) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \pi_0 = \begin{matrix} & u & d & l & r \\ \begin{matrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{matrix} \end{matrix}$$

discount factor = 0.9

		H	
			G

$V(s) = \sum_a \pi(s,a) \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V(s')]$

$V(s) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow V(s) = \begin{bmatrix} 0 \\ -2.5 \\ -2.5 \\ 0 \\ -2.5 \\ 2.5 \\ 0 \\ 0 \\ 2.5 \\ 0 \\ 2.5 \end{bmatrix}$

$V(0) = 0$   
 $V(1) = 1/4 \times 1 \times -10 = -2.5$   
 $V(2) = 1/4 \times 1 \times -10 = -2.5$   
 $V(3) = 0$   
 $V(4) = 1/4 \times 1 \times -10 = -2.5$   
 $V(5) = 1/4 \times 1 \times 10 = 2.5$   
 $V(6) = 0$   
 $V(7) = 0$   
 $V(8) = 1/4 \times 1 \times 10 = 2.5$

		H	
			G

$$V(s) = \sum_{\alpha} \pi(s, \alpha) \sum_{s'} P_{ss'}^{\alpha} [R_{ss'}^{\alpha} + \gamma V(s')]$$

$$V(s) = \begin{bmatrix} 0 \\ -2.5 \\ -2.5 \\ 0 \\ -2.5 \\ +2.5 \\ 0 \\ 0 \\ +2.5 \end{bmatrix} \Rightarrow V(s) = \begin{bmatrix} -0.5625 \\ -3.0625 \\ -3.0625 \\ -0.5625 \\ -1.375 \\ +1.9375 \\ +0.5625 \\ +2.5 \end{bmatrix}$$

$V(0) = \frac{1}{4} \times 1 \times 0.9 \times -2.5 = -0.5625$   
 $V(1) = \frac{1}{4} \times 1 \times -2.5 \times 0.9 + \frac{1}{4} \times 1 \times -10 = -3.0625$   
 $V(2) = \frac{1}{4} \times 1 \times -2.5 \times 0.9 \times 2 + \frac{1}{4} \times 1 \times 2.5 \times 0.9 + \frac{1}{4} \times 1 \times -10 = -3.0625$   
 $V(3) = \frac{1}{4} \times 1 \times 0.9 \times -2.5 = -0.5625$   
 $V(4) = 2 \times \frac{1}{4} \times 0.9 \times 2.5 + \frac{1}{4} \times -10 = -1.375$   
 $V(5) = 2 \times \frac{1}{4} \times 0.9 \times -2.5 + \frac{1}{4} \times 0.9 \times 2.5 + \frac{1}{4} \times 1 \times 10 = 1.9375$   
 $V(6) = 0$   
 $V(7) = \frac{1}{4} \times 0.9 \times 2.5 = 0.5625$   
 $V(8) = \frac{1}{4} \times 10 + \frac{1}{4} \times -2.5 + \frac{1}{4} \times 2.5 = 2.5$

دو ایتريشن از ارزیابی سیاست را گذرانده‌ایم و همچنان همگرا نشده‌ایم این کار را تکرار می‌کنیم تا تقریباً همگرا شویم.

-0.5625	-3.0625	H	-3.0625
	-0.5625	-1.375	+1.9375
0	+0.5625	+2.5	G

-1.07	-3.44	H	-3.44
	-1	-1.63	+1.94
0.13	0.56	2.88	G

-1.5	-3.74	H	-3.61
	-1.53	-2.51	1.8
0.21	0.58	2.9	G

-1.85	-4	H	-3.71
	-1.62	-1.79	1.5
0.27	0.48	2.72	G

-2.14	-4.18	H	-3.83
	-1.56	-1.9	1.6
0.29	0.41	2.81	G

-2.38	-4.27	H	-3.86
	-1.62	-1.85	1.57
0.29	0.44	2.8	G

پس از چندین ایتريشن تقریباً همگرا شده‌ایم. برای سرعت بالاتر تنها را 0.15 در نظر می‌گیریم و به سراغ بهبود سیاست می‌رویم.

-2.38	-4.27	H	-3.86
<del>0.29</del>	-1.62	-1.85	1.57
0.29	0.44	2.8	G

<del>↕</del>	↕	H	↕
<del>↕</del>	↕	↕	↕
↕	↕	↕	G

$\pi(o):$ 

- up:  $0.9 \times -2.38 + 0.25$
- down:  $0.9 \times -4.27 + 0.25$
- right:  $0.9 \times -1.62 + 0.25$
- left:  $0.9 \times -2.38 + 0.25$

 $\Rightarrow$

به همین ترتیب مابقی سیاست ها مشخص می شوند \*

↕	↓	H	↓
<del>↕</del>	↓	↓	↓
→	→	→	G

یک مرحله بهبود سیاست نیز شکل گرفت حال این مراحل را با سرعت بیشتری دنبال می کنیم.

↕	↓	H	↓
<del>↕</del>	↓	↓	↓
→	→	→	G

-2.38	-4.27	H	-3.86
<del>0.29</del>	-1.62	-1.85	1.57
0.29	0.44	2.8	G

0	7.29	H	9
<del>8.1</del>	8.1	9	10
8.1	9	10	G

→	↓	H	↓
<del>→</del>	↘	↘	↓
→	→	→	G

6.56	7.29	H	9
<del>8.1</del>	8.1	9	10
8.1	9	10	G

→	↓	H	↓
<del>→</del>	↘	↘	↓
→	→	→	G

مشاهده می کنیم تنها با سه مرحله اجرای الگوریتم به مقادیر نهایی همگرا می شویم و ارزش استیت ها و سیاست نهایی در تصویر بالا نمایش داده شده است. مشاهده می کنیم زمانی که discount factor نابرابر با صفر داریم مقادیر به گونه ای تغییر می کنند که استیت های کناری هدف دارای بیشترین ارزش و این ارزش یکی یکی به استیت های کناری منتقل می شود. همچنین نسبت به قسمت قبل به سیاست بهتری برای رسیدن به هدف پیدا کرده ایم.

### سوال 3

در این بخش با discount factor برابر 0.9 مسئله را به کمک value iteration حل می کنیم. الگوریتم آن در تصویر زیر نشان داده شده است.

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

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

Loop:  
|  $\Delta \leftarrow 0$   
| Loop for each  $s \in S$ :  
|     $v \leftarrow V(s)$   
|     $V(s) \leftarrow \max_a \sum_{s',r} p(s',r|s,a)[r + \gamma V(s')]$   
|     $\Delta \leftarrow \max(\Delta, |v - V(s)|)$   
until  $\Delta < \theta$

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

0	0	H	0
<del>X</del>	0	0	0
0	0	0	G

0	0	H	0
<del>X</del>	0	0	10
0	0	10	G

0	0	H	9
<del>X</del>	0	9	10
0	9	10	G

0	0	H	9
<del>X</del>	8.1	9	10
8.1	9	10	G

0	7.29	H	9
<del>X</del>	8.1	9	10
8.1	9	10	G

6.56	7.29	H	9
<del>X</del>	8.1	9	10
8.1	9	10	G

6.56	7.29	H	9
<del>X</del>	8.1	9	10
8.1	9	10	G

→	↓	H	↓
<del>X</del>	↘	↘	↓
→	→	→	G

در value iteration کافیت که مقادیر یک یک به سمت خانه های دیگر منتقل شود. ارزش استیت ها نیز مشاهده می کنیم که از استیت های کنار ترمینال استیت منتقل شده تا به خانه های دیگر برود.

تفاوت این دو الگوریتم این است که در policy iteration هر سیاست ارزیابی می شود و سپس آن سیاست بهبود پیدا می کند و این الگوریتم تا جایی که سیاست دچار تغییر شود ادامه پیدا می کند. اما در value iteration تا یک دقت مشخصی فقط ارزش خانه ها را بروزرسانی می کنیم و در آخر سیاست متناظر با آن را پیدا می کنیم. قابل ذکر است که برای ارزش های بدست آمده در هر مرحله می توان سیاستی پیدا کرد اما این کار را انجام نمی دهیم زیرا می دانیم هنوز مقادیر دقیق و درست نیستند و برای پیدا کردن سیاست بهینه باید ارزش خانه ها کامل آپدیت شود پس صبر می کنیم تا این اتفاق بیوفتد و سپس سیاست متناظر آن ارزش ها را پیدا می کنیم.

نکته قابل توجه این است که ارزش های بدست آمده در policy iteration حاصل از یک سیاست مشخص است اما در value iteration ما ارزش خانه ها را از یک سیاست قبلی بدست نمی آوریم بلکه سعی بر پیدا کردن ارزش ماکسیمم هر خانه هستیم. سپس سیاست متناظر آن ارزش های ماکسیمم را بدست می آوریم.

## سوال 4

در هر دوی value iteration و policy iteration امکان اینکه سریع تر به سیاست بهینه همگرا بشویم وجود دارد و خیلی مربوط به مقدار تتا و مقادیر اولیه دارد یعنی اگر فاصله مقادیر اولیه با مقدار واقعی ارزش ها کم باشد به طوری که سریع آن فاصله از تتا کمتر می شود با تعداد گام کمتری همگرا می شویم. اما دقت شود ممکن است با اینکه فاصله ها کم است اما به دلیل تتا خیلی کوچک تغییری در گام ها ایجاد نشود. اگر منظور از نرخ همگرایی مقدار تغییر ارزش هر خانه در هر مرحله است خب مشخصاً تغییرات ارزش ها کمتر از قبل خواهد بود مخصوصاً در گام های ابتدایی در نتیجه نرخ همگرایی کمتر است یعنی مقدار ارزش ها کمتر تغییر می کنند. (راستش لفظاً نمی دانم که توضیحات بالا یعنی نرخ کمتر یا بیشتر)



### سوال 3 - سوال پیاده سازی

#### سوال 1

احتمال لغزش: 0.94 ، discount factor: 0.9

با تابع داده شده make\_map ابتدا دریاچه متناظر با شماره دانشجویی خود را می‌سازیم.

```
array([[0.e+00, 1.e-04, 1.e-04, 1.e+00, 1.e+00, 1.e+00],
       [1.e+00, 1.e+00, 1.e-04, 1.e+00, 1.e+00, 1.e+00],
       [1.e+00, 1.e+00, 1.e-04, 1.e-04, 1.e-04, 1.e+00],
       [1.e+00, 1.e+00, 1.e+00, 1.e+00, 1.e-04, 1.e-04],
       [1.e+00, 1.e+00, 1.e+00, 1.e+00, 1.e+00, 1.e-04],
       [1.e+00, 1.e+00, 1.e+00, 1.e+00, 1.e+00, 0.e+00]])
```

شکل 1 - نقشه دریاچه و مسیر امن

ابتدا **policy iteration** را اجرا می‌کنیم. نتایج به صورت زیر محاسبه شده است.

سیاست بهینه به صورت زیر محاسبه شده است.

```
['r', 'r', 'd', 'T', 'T', 'T']
['T', 'T', 'd', 'T', 'T', 'T']
['T', 'T', 'r', 'r', 'd', 'T']
['T', 'T', 'T', 'T', 'r', 'd']
['T', 'T', 'T', 'T', 'T', 'd']
['T', 'T', 'T', 'T', 'T', 'G']
```

شکل 2 - سیاست بهینه به کمک policy iteration

ارزش های استیت ها به صورت زیر محاسبه شده است.

```
[[19.30392701 23.35221698 28.0764827 0. 0. 0.]
 [ 0. 0. 33.47586961 0. 0. 0.]
 [ 0. 0. 40.58080014 48.86418261 58.50424611 0.]
 [ 0. 0. 0. 0. 69.72287804 82.77856455]
 [ 0. 0. 0. 0. 0. 96.00145616]
 [ 0. 0. 0. 0. 0. 0.]]
```

شکل 3 - ارزش استیت ها به کمک policy iteration

همانطور که مشاهده می‌کنیم ارزش خانه نزدیک خانه هدف بیشترین مقدار را دارد و به همین ترتیب این مقدار با فاصله گرفتن از آن کمتر و کمتر می‌شود. زیرا عواملی همچون خانه های کناری یخ زده و discount روی این موضوع موثر هستند.

```
array([[19.30392701, 0.          , 23.35221698, 19.30392701],
       [19.30392701, 0.          , 28.0764827 , 23.35221698],
       [23.35221698, 33.47586961, 0.          , 28.0764827 ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 40.58080014, 0.          , 28.0764827 ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 48.86418261, 33.47586961],
       [40.58080014, 0.          , 58.50424611, 0.          ],
       [48.86418261, 69.72287804, 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 82.77856455, 58.50424611],
       [69.72287804, 96.00145616, 82.77856455, 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       ...,
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 1.]])
```

شکل 4- بخشی از ارزش استیت اکشن به کمک policy iteration

حال **value iteration** را اجرا می‌کنیم. نتایج به صورت زیر محاسبه شده است.

سیاست بهینه به صورت زیر محاسبه شده است.

```
['r', 'r', 'd', 'T', 'T', 'T']
['T', 'T', 'd', 'T', 'T', 'T']
['T', 'T', 'r', 'r', 'd', 'T']
['T', 'T', 'T', 'T', 'r', 'd']
['T', 'T', 'T', 'T', 'T', 'd']
['T', 'T', 'T', 'T', 'T', 'G']
```

شکل 5 - سیاست بهینه به کمک value iteration

ارزش های استیت ها به صورت زیر محاسبه شده است.

```
[[19.34820256 23.39540585 28.11688762 0. 0. 0. ]
 [ 0. 0. 33.51128881 0. 0. 0. ]
 [ 0. 0. 40.62180714 48.91190064 58.55977791 0. ]
 [ 0. 0. 0. 0. 69.78750319 82.85377208]
 [ 0. 0. 0. 0. 0. 96.06041741]
 [ 0. 0. 0. 0. 0. 0. ]]
```

شکل 6 - ارزش استیت ها به کمک value iteration

```

array([[19.34820256, 0.          , 23.39540585, 19.34820256],
       [19.34820256, 0.          , 28.11688762, 23.39540585],
       [23.39540585, 33.51128881, 0.          , 28.11688762],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 40.62180714, 0.          , 28.11688762],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 48.91190064, 33.51128881],
       [40.62180714, 0.          , 58.55977791, 0.          ],
       [48.91190064, 69.78750319, 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 82.85377208, 58.55977791],
       [69.78750319, 96.06041741, 82.85377208, 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       ...
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ],
       [ 0.          , 0.          , 0.          , 0.          ]])

```

شکل 7- ارزش استیت اکشن ها به کمک value iteration

## سوال 2

احتمال لغزش: 0.7 ، discount factor: 0.9 ، احتمال شکستن خانه‌ها بین 0 و 1 و در مسیر امن احتمال شکستن‌ها برابر 0.001 است.

با تابع داده شده make\_map\_part2 ابتدا دریاچه متناظر با شماره دانشجویی خود را می‌سازیم.

```
array([[0. , 0.001, 0.001, 0.996, 0.772, 0.171],
       [0.67 , 0.375, 0.001, 0.064, 0.976, 0.853],
       [0.335, 0.444, 0.001, 0.001, 0.001, 0.137],
       [0.454, 0.407, 0.807, 0.027, 0.001, 0.001],
       [0.208, 0.175, 0.169, 0.359, 0.463, 0.001],
       [0.775, 0.888, 0.939, 0.853, 0.995, 0. ]])
```

شکل 8 - نقشه دریاچه

ابتدا **policy iteration** را اجرا می‌کنیم. نتایج به صورت زیر محاسبه شده است.

سیاست بهینه به صورت زیر محاسبه شده است.

```
['r', 'r', 'd', 'd', 'd', 'd']
['r', 'r', 'd', 'd', 'd', 'd']
['r', 'r', 'r', 'r', 'd', 'd']
['r', 'r', 'r', 'r', 'r', 'd']
['r', 'r', 'r', 'r', 'r', 'd']
['u', 'u', 'r', 'r', 'r', 'G']
```

شکل 9 - سیاست بهینه به کمک policy iteration

ارزش‌های استیت‌ها به صورت زیر محاسبه شده است.

```
[[ 7.57250441 11.18767623 15.13268487 18.09085162 17.06240108 22.00375189]
 [ 6.96097647 14.37643218 20.71792736 26.36715707 33.09371449 38.33864416]
 [ 9.2451923  18.88845612 27.1294995  35.99545644 44.85899868 53.86404989]
 [10.31733217 18.70357511 33.56148044 44.54449507 56.82811871 69.75017725]
 [13.14222629 22.00859663 34.07528878 50.43333369 69.86419941 88.00316081]
 [ 5.37391463 12.17646357 25.44743617 46.72057938 77.7133737  0.          ]]
```

شکل 10 - ارزش استیت‌ها به کمک policy iteration

```

array([[ 7.57250441,  6.96097647, 11.18767623,  7.57250441],
       [ 7.57250441, 14.37643218, 15.13268487, 11.18767623],
       [11.18767623, 20.71792736, 18.09085162, 15.13268487],
       [15.13268487, 26.36715707, 17.06240108, 18.09085162],
       [18.09085162, 33.09371449, 22.00375189, 17.06240108],
       [17.06240108, 38.33864416, 22.00375189, 22.00375189],
       [ 6.96097647,  9.2451923 , 14.37643218,  7.57250441],
       [ 6.96097647, 18.88845612, 20.71792736, 11.18767623],
       [14.37643218, 27.1294995 , 26.36715707, 15.13268487],
       [20.71792736, 35.99545644, 33.09371449, 18.09085162],
       [26.36715707, 44.85899868, 38.33864416, 17.06240108],
       [33.09371449, 53.86404989, 38.33864416, 22.00375189],
       [ 9.2451923 , 10.31733217, 18.88845612,  6.96097647],
       [ 9.2451923 , 18.70357511, 27.1294995 , 14.37643218],
       [18.88845612, 33.56148044, 35.99545644, 20.71792736],
       [27.1294995 , 44.54449507, 44.85899868, 26.36715707],
       [35.99545644, 56.82811871, 53.86404989, 33.09371449],
       [44.85899868, 69.75017725, 53.86404989, 38.33864416],
       [10.31733217, 13.14222629, 18.70357511,  9.2451923 ],
       [10.31733217, 22.00859663, 33.56148044, 18.88845612],
       [18.70357511, 34.07528878, 44.54449507, 27.1294995 ],
       [33.56148044, 50.43333369, 56.82811871, 35.99545644],
       [44.54449507, 69.86419941, 69.75017725, 44.85899868],
       [56.82811871, 88.00316081, 69.75017725, 53.86404989],
       [13.14222629,  5.37391463, 22.00859663, 10.31733217],
       ...
       [ 5.37391463, 12.17646357, 25.44743617, 22.00859663],
       [12.17646357, 25.44743617, 46.72057938, 34.07528878],
       [25.44743617, 46.72057938, 77.7133737 , 50.43333369],
       [46.72057938, 77.7133737 ,  0.          , 69.86419941],

```

شکل 11 - ارزش استیت اکشن ها به کمک policy iteration

حال **value iteration** را اجرا می‌کنیم. نتایج به صورت زیر محاسبه شده است.

سیاست بهینه به صورت زیر محاسبه شده است.

```
['r', 'r', 'd', 'd', 'd', 'd']
['r', 'r', 'd', 'd', 'd', 'd']
['r', 'r', 'r', 'r', 'd', 'd']
['r', 'r', 'r', 'r', 'r', 'd']
['r', 'r', 'r', 'r', 'r', 'd']
['u', 'r', 'r', 'r', 'r', 'G']
```

شکل 12 – سیاست بهینه به کمک value iteration

ارزش‌های استیت‌ها به صورت زیر محاسبه شده است.

```
[[ 8.68670814 12.40207016 16.43107398 19.51559256 18.87407697 25.04680121]
 [ 7.98660308 15.52954425 22.03799217 27.89791596 34.97229934 41.3486654 ]
 [10.42943158 20.22034083 28.65594644 37.76474822 46.93338577 56.94201728]
 [11.94248964 20.47981599 35.48501098 46.62537777 59.15996181 72.51388207]
 [15.95645006 25.32285147 37.57898426 53.53852888 72.27236376 89.79365522]
 [ 9.00390345 19.88518493 37.4032345 58.78101963 86.50025633 0.      ]]
```

شکل 13 – ارزش استیت‌ها به کمک value iteration

```

array([[ 8.68670814,  7.98660308, 12.40207016,  8.68670814],
       [ 8.68670814, 15.52954425, 16.43107398, 12.40207016],
       [12.40207016, 22.03799217, 19.51559256, 16.43107398],
       [16.43107398, 27.89791596, 18.87407697, 19.51559256],
       [19.51559256, 34.97229934, 25.04680121, 18.87407697],
       [18.87407697, 41.3486654 , 25.04680121, 25.04680121],
       [ 7.98660308, 10.42943158, 15.52954425,  8.68670814],
       [ 7.98660308, 20.22034083, 22.03799217, 12.40207016],
       [15.52954425, 28.65594644, 27.89791596, 16.43107398],
       [22.03799217, 37.76474822, 34.97229934, 19.51559256],
       [27.89791596, 46.93338577, 41.3486654 , 18.87407697],
       [34.97229934, 56.94201728, 41.3486654 , 25.04680121],
       [10.42943158, 11.94248964, 20.22034083,  7.98660308],
       [10.42943158, 20.47981599, 28.65594644, 15.52954425],
       [20.22034083, 35.48501098, 37.76474822, 22.03799217],
       [28.65594644, 46.62537777, 46.93338577, 27.89791596],
       [37.76474822, 59.15996181, 56.94201728, 34.97229934],
       [46.93338577, 72.51388207, 56.94201728, 41.3486654 ],
       [11.94248964, 15.95645006, 20.47981599, 10.42943158],
       [11.94248964, 25.32285147, 35.48501098, 20.22034083],
       [20.47981599, 37.57898426, 46.62537777, 28.65594644],
       [35.48501098, 53.53852888, 59.15996181, 37.76474822],
       [46.62537777, 72.27236376, 72.51388207, 46.93338577],
       [59.15996181, 89.79365522, 72.51388207, 56.94201728],
       [15.95645006,  9.00390345, 25.32285147, 11.94248964],
       ...
       [ 9.00390345, 19.88518493, 37.4032345 , 25.32285147],
       [19.88518493, 37.4032345 , 58.78101963, 37.57898426],
       [37.4032345 , 58.78101963, 86.50025633, 53.53852888],
       [58.78101963, 86.50025633,  0.          , 72.27236376],

```

شکل 14- ارزش استیت اکشن ها به کمک value iteration



## تحلیل:

اولین نکته‌ای که قابل ذکر است این است که در بخش اول همه استیت‌های غیر از مسیر امن ترمینال استیت تلقی می‌شدند که منجر می‌شد فقط همان استیت‌های مسیر امن دارای ارزش شوند؛ اما در بخش دوم همه استیت‌ها دارای ارزش هستند زیرا یک احتمال برای نشکستن آن‌ها وجود دارد. متناظراً این موضوع برای سیاست نیز برقرار است که سیاست فقط برای خانه‌های غیر از ترمینال استیت مفهوم پیدا می‌کند.

اندازه یا مقدار ارزش استیت‌های داخل مسیر امن در بخش دوم کاهش پیدا کرده است زیرا مقدار مجازات افتادن در آب از استیت‌های خیلی بیشتری به صورت discounted روی آن‌ها اثر می‌گذارد در صورتی که در بخش اول تنها استیت‌های کنار مسیر امن تاثیرگذار بودند.

نکته دیگر این است که لزوماً فاصله از خانه هدف باعث کاهش ارزش نمی‌شود بلکه فاصله از مسیر امن تاثیرگذارتر است زیرا با اینکه می‌توان از خانه هدف دور بود اما با قرار داشتن روی مسیر امن یا نزدیکی آن با سیاست بهینه با احتمال بالاتری به سلامت به خانه هدف می‌رسیم.

سیاست بهینه نیز در بخش اول دقیقاً مطابق نقشه بدست آمده است و در بخش دوم نیز سیاست بهینه برای استیت‌های غیر از مسیر امن به گونه‌ای بدست آمده است که با کمترین فاصله و هزینه به مسیر امن بپیوندند.

مشخصاً سرعت همگرایی در بخش اول بالاتر است زیرا هم تعداد استیت‌هایی که نیاز به محاسبه ارزش دارند بیشتر است و هم استیت‌های خیلی بیشتری بر روی هر استیت تاثیر می‌گذارند.

### سوال 3

الف) تاثیر مقدار ترشولد

Value Iteration with  $\theta = 2$

```
['u', 'l', 'l', 'l', 'l', 'l', 'l', 'l', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r']
['u', 'u', 'u', 'u', 'l', 'd', 'd', 'l', 'd', 'r', 'u', 'u', 'u', 'u', 'u']
['u', 'u', 'r', 'u', 'l', 'l', 'l', 'l', 'd', 'r', 'r', 'l', 'r', 'u', 'u']
['r', 'u', 'u', 'u', 'u', 'u', 'd', 'l', 'l', 'd', 'u', 'u', 'r', 'u', 'l']
['u', 'u', 'u', 'r', 'r', 'r', 'l', 'l', 'l', 'l', 'l', 'r', 'r', 'u', 'l']
['u', 'u', 'u', 'r', 'r', 'u', 'u', 'l', 'd', 'd', 'd', 'r', 'd', 'u', 'd']
['u', 'd', 'd', 'r', 'u', 'u', 'u', 'r', 'd', 'd', 'd', 'd', 'd', 'r', 'd']
['r', 'r', 'r', 'd', 'u', 'l', 'u', 'r', 'r', 'd', 'd', 'd', 'd', 'd', 'd']
['r', 'r', 'r', 'u', 'l', 'l', 'd', 'u', 'r', 'r', 'r', 'd', 'd', 'd', 'd']
['d', 'd', 'd', 'd', 'd', 'd', 'r', 'd', 'r', 'r', 'r', 'd', 'd', 'r', 'd']
['d', 'r', 'd', 'd', 'l', 'l', 'r', 'r', 'd', 'r', 'r', 'd', 'd', 'd', 'd']
['d', 'd', 'd', 'd', 'd', 'd', 'r', 'r', 'r', 'r', 'r', 'd', 'd', 'd', 'd']
['d', 'l', 'l', 'l', 'l', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'd', 'd', 'd']
['d', 'l', 'l', 'u', 'u', 'r', 'd', 'r', 'r', 'r', 'r', 'r', 'r', 'd', 'd']
['u', 'u', 'l', 'l', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'G']
```

شکل 15- سیاست نهایی با  $\theta = 2$

Value Iteration with  $\theta = 0.000001$

```
['u', 'l', 'l', 'l', 'l', 'l', 'l', 'l', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r']
['u', 'u', 'u', 'u', 'l', 'd', 'd', 'l', 'd', 'r', 'u', 'u', 'u', 'u', 'u']
['u', 'u', 'r', 'u', 'l', 'l', 'l', 'l', 'd', 'r', 'r', 'l', 'r', 'u', 'u']
['r', 'u', 'u', 'u', 'u', 'u', 'l', 'd', 'd', 'd', 'u', 'd', 'r', 'u', 'l']
['u', 'u', 'u', 'u', 'r', 'u', 'd', 'l', 'l', 'd', 'l', 'r', 'd', 'u', 'd']
['u', 'u', 'u', 'r', 'r', 'r', 'd', 'd', 'd', 'd', 'd', 'r', 'd', 'l', 'd']
['u', 'u', 'd', 'r', 'u', 'r', 'r', 'r', 'd', 'd', 'd', 'd', 'd', 'r', 'd']
['r', 'r', 'r', 'd', 'u', 'r', 'r', 'r', 'r', 'd', 'd', 'd', 'd', 'd', 'd']
['r', 'r', 'r', 'u', 'l', 'd', 'r', 'u', 'r', 'r', 'r', 'd', 'd', 'd', 'd']
['d', 'd', 'd', 'd', 'd', 'r', 'r', 'd', 'r', 'r', 'r', 'd', 'd', 'r', 'd']
['d', 'r', 'd', 'd', 'l', 'd', 'r', 'r', 'd', 'r', 'r', 'd', 'd', 'd', 'd']
['d', 'd', 'd', 'd', 'd', 'd', 'r', 'r', 'r', 'r', 'r', 'd', 'd', 'd', 'd']
['d', 'l', 'l', 'l', 'l', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'd', 'd', 'd']
['d', 'l', 'l', 'u', 'd', 'r', 'd', 'r', 'r', 'r', 'r', 'r', 'r', 'd', 'd']
['u', 'u', 'l', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'G']
```

شکل 16- سیاست نهایی با  $\theta = 0.000001$

همانطور که مشاهده می‌فرمایید هر دو به یک سیاست همگرا شده‌اند و دلیل این موضوع این است که تا اگر مقدار معقولی باشد (در اردر خود پاداش ها یا بیشتر نباشد) تنها به دقت تخمین ما نسبت به ارزش استیت ها مربوط می‌شود و تاثیری در سیاست بهینه ندارد. در نتیجه تنها کافیت تا مقداری باشد که از اردر پاداش های دریافتی کوچک باشد برای مثال در این سوال که پاداش ها برابر 10 و 100 هستند اگر تا نزدیک 1 باشد مناسب خواهد بود. و هر چقدر ما این تا را کاهش دهیم صرفا تخمین خود را ارزش ها را دقیق تر کرده‌ایم.

### ب) تاثیر مقدار پاداش خانه هدف

همانطور که مشاهده می‌کنید در دو حالت بالا به سیاستی که منجر به رسیدن به خانه هدف باشد نرسیدیم. حال با تغییر مقدار پاداش خانه هدف به 10000 این موضوع را بررسی می‌کنیم.

```
[ 'r', 'r', 'r', 'r', 'r', 'd', 'r', 'r', 'r', 'r', 'd', 'd', 'd', 'd', 'd', 'd' ]
[ 'r', 'd', 'r', 'd', 'd', 'd', 'd', 'r', 'd', 'r', 'd', 'd', 'd', 'd', 'd', 'd' ]
[ 'r', 'r', 'r', 'r', 'r', 'd', 'd', 'd', 'd', 'd', 'd', 'd', 'd', 'd', 'd', 'd' ]
[ 'r', 'r', 'r', 'r', 'r', 'd', 'd', 'd', 'd', 'd', 'd', 'd', 'd', 'd', 'd', 'd' ]
[ 'r', 'r', 'r', 'r', 'r', 'r', 'd', 'd', 'd', 'd', 'd', 'd', 'd', 'd', 'd', 'd' ]
[ 'd', 'r', 'r', 'r', 'r', 'r', 'd', 'd', 'd', 'd', 'd', 'd', 'd', 'd', 'd', 'd' ]
[ 'd', 'd', 'd', 'r', 'r', 'r', 'r', 'r', 'd', 'd', 'd', 'd', 'd', 'd', 'd', 'd' ]
[ 'd', 'r', 'r', 'd', 'r', 'r', 'r', 'r', 'r', 'd', 'd', 'd', 'd', 'd', 'd', 'd' ]
[ 'd', 'r', 'r', 'r', 'd', 'd', 'd', 'r', 'r', 'r', 'r', 'd', 'd', 'd', 'd', 'd' ]
[ 'd', 'd', 'd', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'd', 'd', 'd', 'd', 'd' ]
[ 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'd', 'r', 'r', 'd', 'd', 'd', 'd', 'd' ]
[ 'd', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'd', 'd', 'd', 'd' ]
[ 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'd', 'd', 'd', 'd' ]
[ 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'd', 'd' ]
[ 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'r', 'G' ]
```

شکل 17 - سیاست نهایی با تغییر پاداش

با توجه به نتیجه بدست آمده در شکل 17 مشاهده می‌کنیم که به سیاست بهینه همگرا شده‌ایم. یعنی به سیاستی که به خانه هدف می‌رسد رسیدیم. دلیل این موضوع این است که اگر پاداش کم باشد خانه های با فاصله از خانه هدف مقدار پاداش مثبت را به نوعی متوجه نمی‌شوند و پاداش های منفی که تعداد خیلی بیشتری نسبت به تک پاداش مثبت را دارند متوجه می‌شوند. پس باید با افزایش پاداش خانه هدف تاثیر پاداش های منفی را کم کنیم و اجازه دهیم این پاداش به خانه های با فاصله منتقل شود.

[1] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.