



لطفاً به نکات زیر توجه کنید:

- مهلت ارسال این تمرین ۹ دی است.

- در صورتی که به اطلاعات بیشتری نیاز دارید می‌توانید به صفحه‌ی تمرین در وبسایت درس مراجعه کنید.

- این تمرین شامل سوال‌های برنامه‌نویسی می‌باشد، بنابراین توجه کنید که حتماً موارد خواسته‌شده در سوال را رعایت کنید. در صورتی که به هر دلیلی سامانه‌ی داوری نتواند آن را اجرا کند مسئولیت آن تنها به عهده‌ی شماست.

- ما همواره هم‌فکری و همکاری را برای حل تمرین‌ها به دانشجویان توصیه می‌کنیم. اما هر فرد باید تمامی سوالات را به تنهایی تمام کند و پاسخ‌ارسالی حتماً باید توسط خود دانشجو نوشته‌شده باشد. لطفاً اگر با کسی هم‌فکری کردید نام او را ذکر کنید. در صورتی که سامانه‌ی تطبیق، تقلبی را تشخیص دهد متأسفانه هیچ مسئولیتی بر عهده‌ی گروه تمرین نخواهد بود.

- لطفاً برای ارسال پاسخ‌های خود از راهنمای موجود در صفحه‌ی تمرین استفاده کنید.

- هر سوالی درباره‌ی این تمرین را می‌توانید در گروه درس مطرح کنید و یا از دستیاران حل تمرین بپرسید.

## سوالات عملی

سوالات عملی این سری مربوط به یادگیری تقویتی میباشند.

برای پیاده‌سازی یادگیری تقویتی از محیط `gym` استفاده میکنیم که یکی از معروفترین محیط‌های توسعه یادگیری تقویتی میباشد.

برای اطلاعات بیشتر و نصب محیط به لینک زیر مراجعه کنید:

<https://gym.openai.com/>

### سوال ۱ (ماشین بازی):

در این سوال به پیاده‌سازی عامل یادگیری تقویتی برای یک ماشین میپردازیم.

برای اینکار از محیط `MountainCar` که به طور پیشفرض در `gym` قرار دارد استفاده میکنیم و شما صرفاً باید هوش عامل یادگیرنده و پارامترهای آن را پیاده‌سازی کنید.

برای شروع پیاده‌سازی فایل `sample.py` را از سایت درس دریافت کنید و متغیر `env` در آن را به مقدار `'MountainCar-v0'` تغییر دهید و آن را اجرا کنید.

برای اطلاعات بیشتر به لینک زیر مراجعه کنید:

<https://gym.openai.com/envs/MountainCar-v0/>

### سوال ۲ (مسئله پاندول):

در این مسئله شما وظیفه دارید با استفاده از روش‌های یادگیری تقویتی سعی کنید تعادل پاندول را حفظ کنید.

پاندول از یک موقعیت رندوم شروع به حرکت کرده و هدف این است که آن را مستقیم و رو به بالا (با زاویه ۹۰ نسبت به محور مختصات) نگه داشت.

مشابه سوال قبلی برای شروع میتوانید فایل `sample.py` را از سایت درس دریافت کنید و متغیر `env` در آن را به مقدار `'Pendulum-v0'` تغییر دهید و آن را اجرا کنید.

برای اطلاعات بیشتر به لینک زیر مراجعه کنید:

<https://gym.openai.com/envs/Pendulum-v0/>



## سوالات تئوری

۱- در محیط مشخص شده زیر  $A$  حالت شروع است و حالت هایی که دور آنها مربع کشیده شده حالات خروجی هستند. در یک حالت خروجی تنها عمل ممکن خروج می‌باشد که منجر به دریافت پاداش و پایان بازی می‌شود. در حالتی که خروجی نباشند میتوان به چپ یا راست رفت. در سوالات زیر فرض کنید که در ابتدا برای همه حالات  $S$  مقدار  $V_S(0)$  برابر صفر است.



در ابتدا فرض کنید که  $\text{discount}$  برابر ۱ است ( $\gamma = 1$ ) و اعمال همیشه به درستی انجام می‌شوند. (همه موارد زیر نیاز به توضیح دارند)

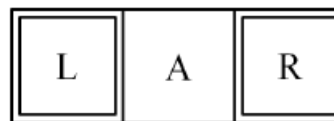
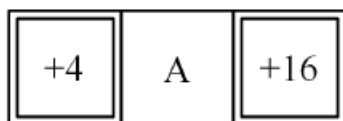
الف) بهینه ترین مقدار  $V^*(A)$  چه خواهد بود؟  
ب) هنگام اجرا  $\text{value-iteration}$ ، در چه مرحله ای  $A$  غیر صفر خواهد بود و مقدار آن چیست؟

ج) بعد از چند مرحله  $V_k(A) = V^*(A)$  خواهد بود؟

د) اگر مقدار  $\text{discount}$  برابر  $0/5$  باشد مقدار بهینه  $A$  چه خواهد بود؟

ه) برای چه بازه ای از مقادیر  $\text{discount}$  رفتن به راست از حالت  $A$  بهینه خواهد بود؟

۲- شکل زیر را در نظر بگیرید



در این سناریو  $\text{discount}$  را برابر ۱ فرض کنید. در اینجا اطلاعاتی از جزئیات  $\text{MDP}$  نداریم بنابراین از یادگیری تقویتی استفاده می‌کنیم.

موارد زیر به ترتیب مشاهده شده‌اند ( $x$  حالت پایانی و به معنای خروج از بازی است):



$s$	$a$	$s'$	$r$
$A$	<i>Right</i>	$R$	0
$R$	<i>Exit</i>	$X$	16
$A$	<i>Left</i>	$L$	0
$L$	<i>Exit</i>	$X$	4
$A$	<i>Right</i>	$R$	0
$R$	<i>Exit</i>	$X$	16
$A$	<i>Left</i>	$L$	0
$L$	<i>Exit</i>	$X$	4

الف) اگر از نرخ یادگیری  $\alpha = 0.5$  استفاده کنیم؛ با استفاده از روش Temporal Difference به چه مقداری برای  $A$  می‌رسیم؟ (در ابتدا مقدار برای همه حالت‌ها ۰ است)

ب) اگر از نرخ یادگیری  $\alpha = 0.5$  استفاده کنیم؛ با استفاده از روش Q-Learning به چه مقداری برای  $Q(A, \text{Right})$  می‌رسیم؟ (در ابتدا مقدار  $Q$  برای همه  $(s, a)$ ‌ها ۰ است)

۳- روباتی در نظر بگیرید که دو حالت OK و HOT دارد و می‌تواند SLOW و یا FAST حرکت کند. جدول احتمالات و جایزه‌ها به صورت زیر است :

$s$	$a$	$s'$	$T(s, a, s')$	$R(s, a, s')$
OK	SLOW	OK	1.0	+1
OK	FAST	OK	0.5	+2
OK	FAST	HOT	0.5	+2
HOT	SLOW	OK	1.0	+1
HOT	FAST	HOT	0.5	+2
HOT	FAST	OK	0.5	-10

با فرض اینکه مقدار discount برابر  $0.8$  است،

الف) روش value-iteration را ۳ مرحله اجرا کنید.

ب) با در نظر گرفتن یک policy رندم، روش policy-iteration را ۳ مرحله اجرا کنید.

ج) دو روش ذکر شده را با هم مقایسه کنید. کدام یک سریع‌تر است؟ چرا؟