



## Adversarial Search

تمرین سوم، بخش دوم

مهلت ارسال: ۱۳ آبان

- مهلت ارسال پاسخ تا ساعت ۲۳:۵۹ روز مشخص شده است.
- این تمرین به علت منتهی شدن به میانترم مهلت تاخیر ندارد.
- هم‌کاری و هم‌فکری شما در انجام تمرین مانعی ندارد اما پاسخ‌های هر کس حتما باید توسط خود او نوشته شده باشد.
- در صورت هم‌فکری و یا استفاده از هر منبع خارج از کتاب و اسلایدهای درس، نام هم‌فکران و آدرس منابع مورد استفاده برای حل سوال مورد نظر را ذکر کنید.
- لطفا تصویری واضح از پاسخ‌های سوالات نظری بارگذاری کنید. در غیر این صورت پاسخ شما تصحیح نخواهد شد.

### سوالات نظری (۴۰ نمره)

۱. (۲۵ نمره) درست یا نادرست بودن عبارت‌های زیر را با ذکر دلیل مشخص کنید. (بازی غیر بهینه برای یک بازیکن به این معنا است که حداقل یک بار شاخه غیر بهینه، شاخه‌ای که بیشترین امتیاز را برای او به ارمغان نمی‌آورد، را در درخت بازی انتخاب کند و همچنین در بازی رندوم احتمال انتخاب هر کدام از شاخه‌های درخت با هم برابر است.)

(آ) در یک بازی دو نفره non zero-sum مقدار به دست آمده برای ریشه درخت بازی، دوتایی  $(U_A, U_B)$  است. در صورتی که بازیکن  $B$  غیر بهینه بازی کند، مقدار امتیاز  $A$  حداقل  $U_A$  خواهد بود.

(ب) در یک بازی دو نفره zero-sum نفر  $A$  مقدار  $minimax$  را برای ریشه درخت بازی عدد  $X$  به دست آورده است. در صورتی که امتیاز تمامی برگ‌ها با هم متفاوت باشند عبارات زیر را مشخص کنید.

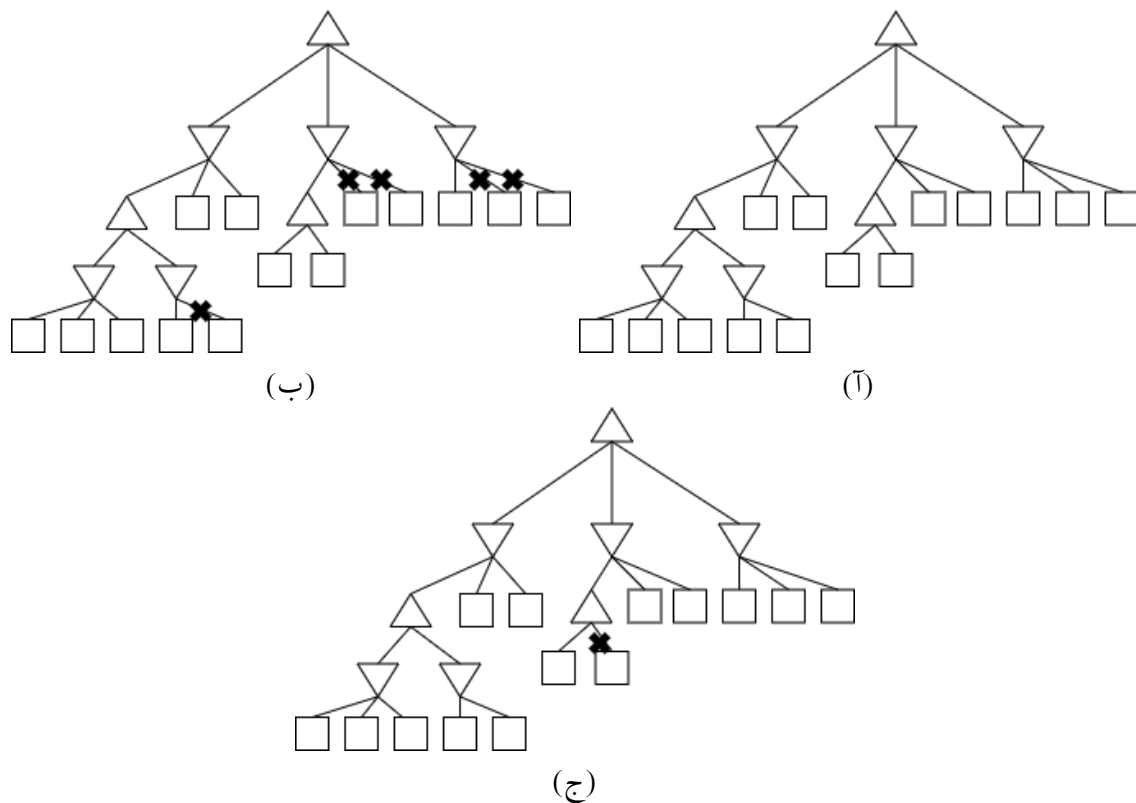
i. اگر نفر  $B$  غیر بهینه بازی کند و  $A$  آن را نداند و با استراتژی  $minimax$  بازی کند، امتیاز نهایی برای  $A$  ممکن است از  $X$  کمتر باشد.

ii. در صورتی که  $B$  رندوم بازی کند،  $A$  استراتژی‌ای دارد که تضمین کند به امتیازی بهتر از  $X$  دست می‌یابد.

iii. در صورتی که  $B$  غیر بهینه بازی کند،  $A$  با استراتژی  $minimax$  حتما به امتیازی بیشتر از  $X$  دست می‌یابد.

iv. در صورتی که  $B$  غیر بهینه بازی کند و  $A$  تمام حرکات  $B$  را از قبل بداند،  $A$  استراتژی‌ای دارد که به امتیازی بهتر از  $X$  دست یابد.

۲. (۱۵ نمره) به شکل ۱ توجه کنید. برای هر یک از موارد خواسته شده مشخص کنید که آیا هرس انجام شده امکان پذیر است یا خیر. در صورت امکان پذیر بودن مثال بیاورید و در صورت عدم امکان توضیح دهید.



شکل ۱: سوال ۲.

## سوالات عملی (۶۵ نمره)

۱. (۶۵ نمره) در این تمرین می‌خواهیم با استفاده از الگوریتم‌هایی که یاد گرفتیم یک حریف برای بازی **isolation** پیاده‌سازی کنیم. این بازی دو نفره روی یک صفحه  $7 \times 7$  انجام می‌شود که هر کدام از دو نفر یک مهره **اسب** دارند که به مانند مهره اسب شطرنج حرکت می‌کند. اسب‌ها در هر خانه‌ای که قرار بگیرند آن خانه را تخریب می‌کنند و در حرکات بعدی، خودشان یا حریفشان در آن خانه نمی‌توانند قرار بگیرند. در نهایت بازیکنی که در نوبت خود نتواند حرکت کند بازنده و بازیکن دیگر برنده است.

شما برای این بازی یک عامل هوشمند با استفاده از درخت **minimax** طراحی می‌کنید و در ادامه هرس آلفابتا را پیاده‌سازی می‌کنید. همچنین قرار است یک تابع ارزیابی پیاده‌سازی کنید و از آن برای ارزیابی حالات میانی کمک بگیرید. به دلیل محدودیت ۱۵۰ میلی ثانیه‌ای بررسی درخت تا انتها قابل انجام نیست. لازم به ذکر است در صورتی که انتخاب حرکت بیشتر از این مقدار طول بکشد یا حرکت انجام شده نامعتبر باشد، بلافاصله بازنده بازی خواهید بود.

## ملاحظات

- حتما پیش از هر چیز فایل `readme.md` را برای درک خواسته‌های پروژه و توضیحات اولیه راجع به پروژه مطالعه نمایید.
- برای ایجاد شهود اولیه چند حریف تمرینی در `sample_players.py` قرار گرفته است.
- توجه کنید که تنها فایلی که شما باید ادیت کرده و آپلود کنید فایل `game_agent.py` است. بنابراین اکیدا از ایجاد تغییر در سایر فایل‌ها خودداری کنید.
- توصیه می‌شود از ابزار مصور سازی که در اختیارتان قرار داده شده است برای بررسی رفتار `agent` خود کمک بگیرید.

- لزومی به نوشتن داکيومنت برای توضیح کد وجود ندارد ولی توصیه می‌شود با کامنت‌های مناسب در کنار کد به توضیح اجمالی رفتار کد (به ویژه تابع ارزیابی) بپردازید. در صورتی که منطق کد شما برای مصحح احراز نشود امتیازی به آن قسمت کد منظور نخواهد شد.
- نحوه نمره دهی:

- ۱۵ نمره برای پیاده سازی تابع `MinimaxPlayer.minimax()`
- ۵ نمره برای `AlphaBetaPlayer.get_move()`
- ۲۰ نمره برای پیاده سازی `AlphaBetaPlayer.alphabeta()`
- ۲۰ نمره برای بهتر بودن نتیجه agent شما از `AB_Improved`
- ۵ نمره امتیازی برای بهترین کدها در میان کد تمامی دانشجویان