

### لطفاً به نکات زیر توجه فرمایید:

- مهلت ارسال این تمرین تا ۱۹ اردیبهشت 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.

- آدرس صفحه‌ی تمرین:

[https://iust-courses.github.io/ai982/assignments/03\\_minimax](https://iust-courses.github.io/ai982/assignments/03_minimax)

- آدرس گروه درس:

<https://groups.google.com/forum/#!forum/ai982>

## سوالات عملی:

در این تمرین به پیاده سازی الگوریتم های مینیمکس و هرس آلفا-بتا بر روی بازی شطرنج میپردازیم. ابتدا فایل زیپ را از سایت درس دانلود کنید و مازول های PyQt5 و Python-Chess را با دستورات زیر نصب کنید.

```
$pip install pyqt5
$pip install python-chess
```

برای اطلاعات بیشتر به لینک زیر مراجعه کنید.

<https://pypi.org/project/python-chess/>

### ۱. مینیمکس ساده (۳۰ نمره)

در فایل `MinimaxAI.py` الگوریتم مینیمکس را برای بازی شطرنج پیاده سازی کنید. کلاس مینیمکس شما در هر مرحله متغیر `board` را به عنوان ورودی دریافت میکند و در متد `choose_move` خروجی الگوریتم را برمیگرداند.

از آنجایی که فضای حالت بازی شطرنج بسیار بزرگ میباشد نمیتوان درخت مینیمکس را تا انتها ادامه داد بنابراین باید برای هر گره در درخت یک تابع هیوریستیک بدست بیاورید و بر اساس آن خروجی بازی را برگردانید.

توجه کنید که با توجه به اینکه عمق درخت کم میباشد ممکن است عامل شما در یک حلقه گیر کند به عنوان مثال یک مهره را حرکت دهد و دوباره به خانه قبلی خود برگرداند که باید از آن جلوگیری کنید.

برای بررسی کد خود در فایل `gui_chess.py` بازیکن اول و بازیکن دوم را از بین عامل خودتان و عامل رندوم انتخاب کنید و فایل `gui_chess.py` را اجرا کنید.

### ۲. هرس آلفا-بتا (۳۰ نمره)

با توجه به قسمت قبل و عمق کم الگوریتم مینیمکس احتیاج به هرس کردن درخت داریم تا بتوانیم عمق را افزایش دهیم.

در فایل `AlphaBetaAI.py` الگوریتم مینیمکس را با هرس آلفا-بتا برای بازی شطرنج پیاده سازی کنید.

همچنین در این قسمت نیز با اینکه درخت را هرس میکنیم باز به عمق خوبی از درخت نمیتوانیم برسیم بنابراین به تابع هیوریستیک احتیاج داریم.

برای بررسی کد خود در فایل `gui_chess.py` یکی از عامل ها را عامل آلفا-بتا و عامل دیگر را

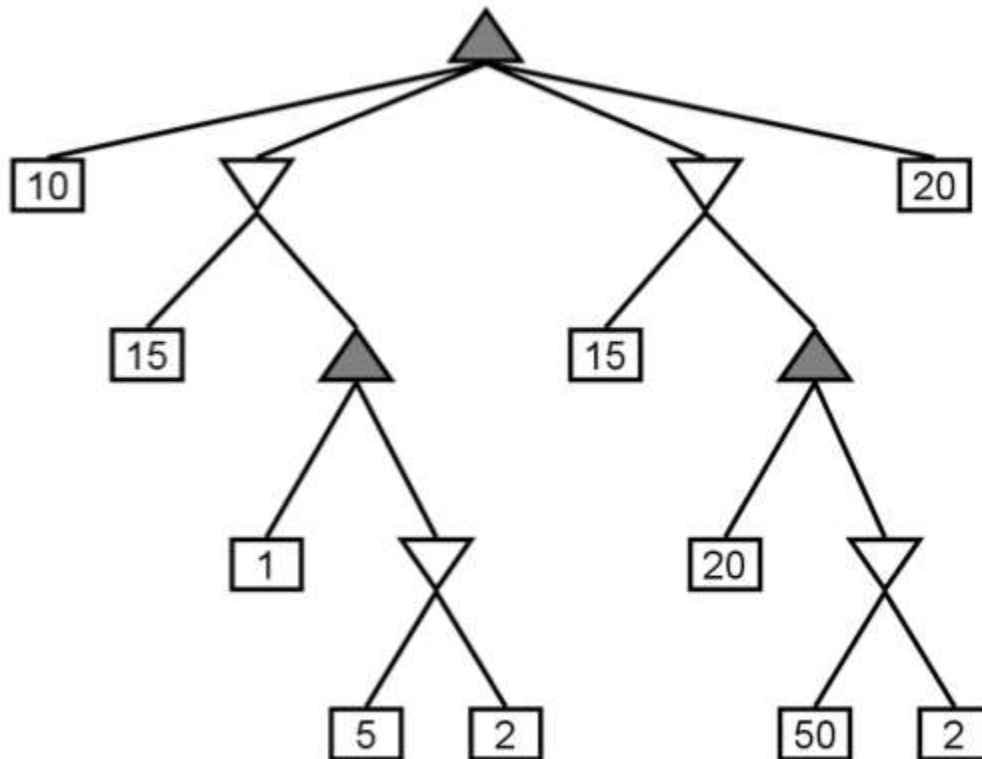
عامل مینیمکس (یا رندم) قرار دهید همچنین سرعت اجرا شدن آنها با یک عمق یکسان را باهم مقایسه کنید.

### ۳. مینیمکس خفن (امتیازی)

با جستجو در رابطه با هیوریستیک های موجود که برای برطرف کردن مشکل عمق درخت مینیمکس در بازی شطرنج پیاده سازی شده اند یکی از آنها را پیاده سازی کنید.

#### سوالات تئوری:

۱. درخت مینیمکس زیر را در نظر بگیرید: (۲۵ نمره)



الف) مقدار minimax برای ریشه درخت (root) چه عددی است؟

ب) نودهایی را که از طریق alpha-beta pruning مورد بازدید قرار نمی گیرند را مشخص کنید. (فرض کنید فرزندان از چپ به راست بازدید میشوند.)

ج) آیا ترتیب دیگری برای نودهای فرزند (children of root) وجود دارد که pruning بیشتری حاصل شود؟ اگر بله، ترتیب را بیان کنید.

د) یک روش کلی برای ترتیب فرزندان ریشه (children of root) بیان کنید که احتمال pruning را افزایش میدهد. لطفا توضیحات مختصر باشد اما حتما توضیحات شما شامل اینکه برای نودها min و max چه راهکاری در نظر گرفته اید، باشد.

۲. مقدار حافظه لازم برای اجرای الگوریتم minimax با روش alpha-beta pruning را برحسب big O بیان کنید. (فاکتورهایی که فکر میکنید به آن مرتبط است را خودتان تعریف کنید)(۱۵ نمره)