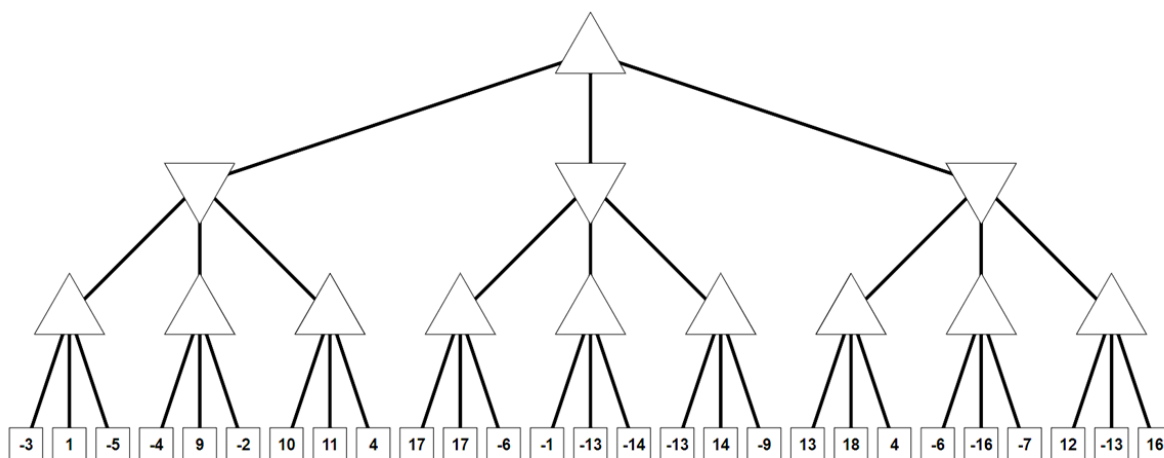
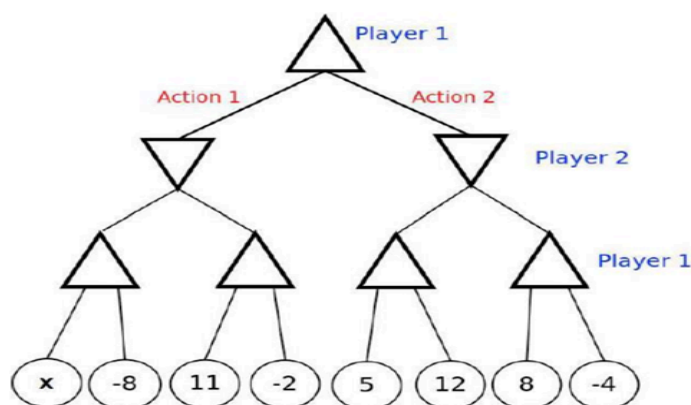
	تمرین سری سوم هوش مصنوعی و سیستم‌های خبره	نام مدرس: دکتر محمدی دستیاران تمرین: ترانه عبداللهی، طاهای بیکلریان، مهرشاد فلاح، نیما سعیدی
		مهلت تحویل: یکشنبه 11م آبان ماه

بخش تئوری (60 نمره):

1- الف) روی درخت زیر هرس آلفا-بتا را انجام دهید و کل عملیات و خروجی نودها را مشخص کنید (12 نمره):



ب) در درخت زیر به ازای چه مقادیری از x ، بازیکن 1 به صورت تضمینی Action 1 را انتخاب می‌کند؟
ج) فرض کنید بازیکن 2 اقدامات را به صورت تصادفی انتخاب می‌کند، به طوری که انتخاب هر اقدام از جانب او احتمال برابری دارد. به ازای چه مقادیری از x ، بازیکن 1 به طور تضمینی Action 1 را انتخاب می‌کند؟



2- فرض کنید K ماشین بخت‌آزمایی داریم که هر کدام 1 سکه می‌گیرند و با یک احتمالی به ما جایزه می‌دهند. (توجه کنید که احتمال هر دستگاه را نمی‌دانیم و آن را فقط با آزمایش می‌توانیم به دست آوریم). می‌خواهیم N سکه اولیه را جوری استفاده کنیم که بیشترین سود را دریافت کنیم. پس تصمیم می‌گیریم از الگوریتم UCB1 استفاده کنیم. (18 نمره) ($C=1$)

الف) فرض کنید $K = 10$ است و همه آن 10 ماشین خراب هستند و همیشه به ما جایزه می‌دهند. صحت این جمله را بررسی کنید: (بعد از پرداخت دقیقاً 20 سکه، هر ماشین 2 بار استفاده شده است).

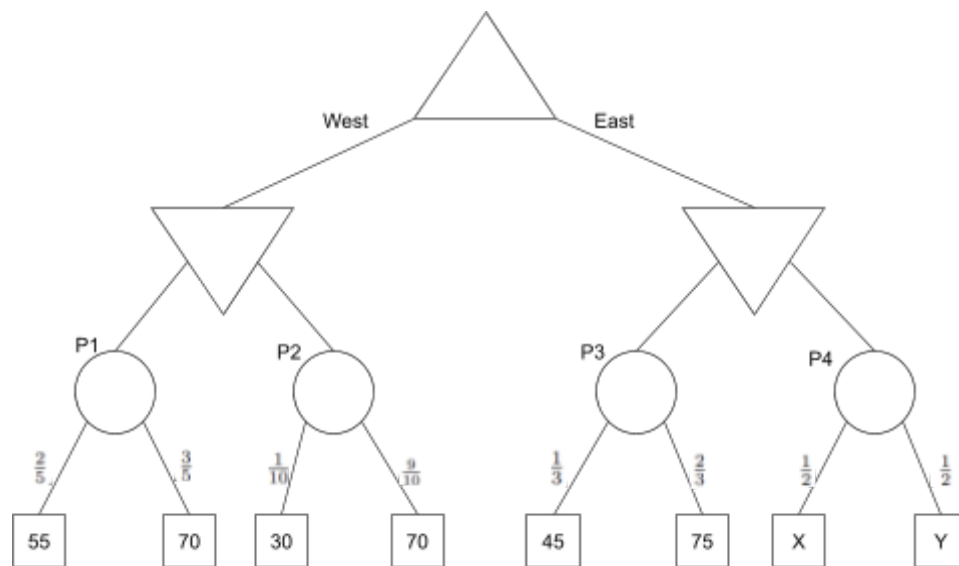
ب) حالا فرض کنید $K = 2$ است و هر ماشین را یک بار تست کرده‌ایم: ماشین اول جایزه نداد و ماشین دوم جایزه داد. اما همان موقع ماشین دوم خراب می‌شود و دیگر قرار نیست جایزه‌ای بدهد بدون آن که ما متوجه شویم. چند سکه دیگر سر ماشین دوم هدر می‌دهیم تا بالاخره ماشین اول را امتحان کنیم؟

3- Pacman در حال انجام یک بازی دشوار است. چهار پورتال به دنیای غذا وجود دارد. با این حال، هر پورتال توسط یک روح محافظت می‌شود. علاوه بر این، نه Pacman و نه روح‌ها نمی‌دانند که پشت هر پورتال چند عدد خوراکی وجود دارد، اما آن‌ها از گزینه‌ها و احتمالات مربوط به همه پورتال‌ها جز آخرین پورتال آگاه هستند. Pacman ابتدا حرکت می‌کند و می‌تواند به سمت شرق یا غرب برود. پس از آن روح می‌تواند یکی از پورتال‌های موجود را مسدود کند. (15 نمره)

درخت بازی زیر را در نظر بگیرید. گره‌های max مربوط به pacman و گره‌های min مربوط به روح‌ها، و گره‌های مربوط به پورتال‌ها گره‌های تصادفی هستند که احتمال مربوط به آن‌ها بر روی یال‌ها مشخص شده است.

در صورت تساوی حرکت سمت چپ انتخاب می‌شود.

فرض کنید pacman و روح‌ها به صورت بهینه بازی می‌کنند.



الف) مقادیر گره‌هایی که به X و Y وابسته نیستند را بدست آورید.

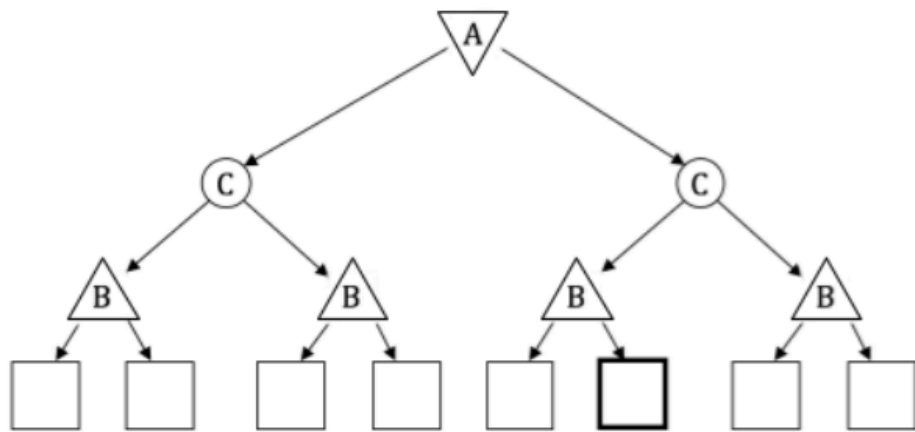
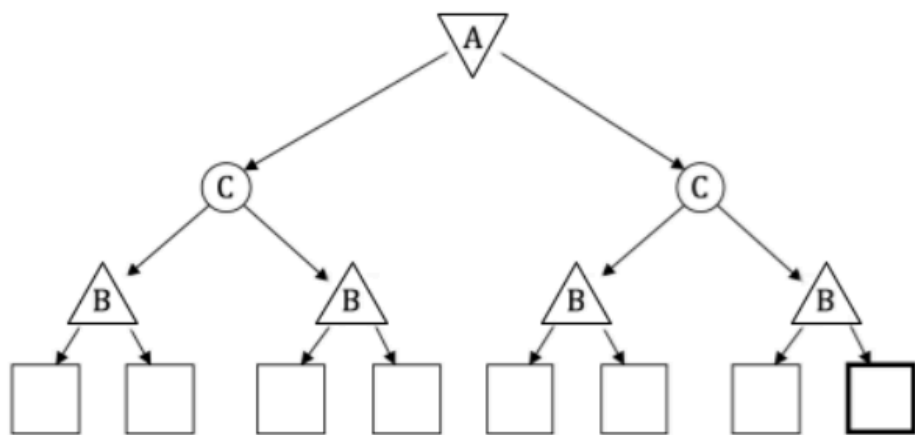
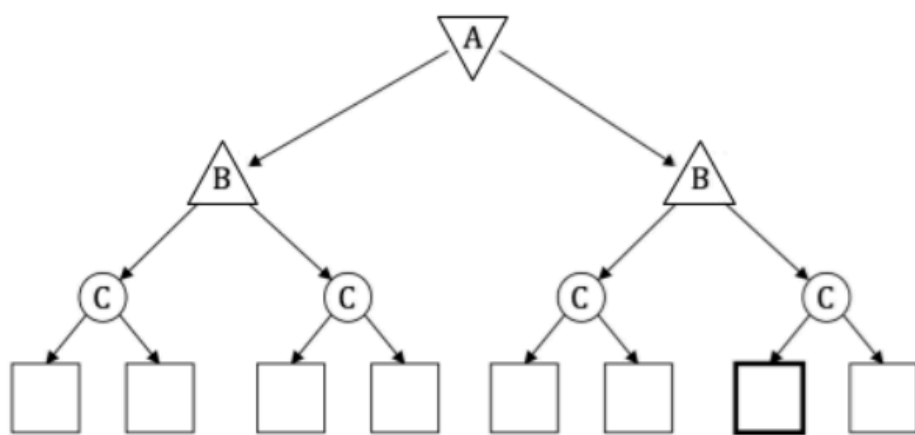
ب) X و Y چه شرایطی را باید ارضا کنند تا pacman به سمت شرق حرکت کند؟ و چه شرایطی لازم است که pacman حتما به پورتال 4 برسد؟

نکته: در نظر داشته باشید که X و Y تعداد پلت‌های غذایی هستند و باید عددهای صحیح در بازی زیر باشند:

$$X, Y \in \{0, 1, 2, 3, \dots\}$$

4- در این سوال، بازیکن A یک min، بازیکن B یک max و C یک گره تصادفی است. تمام فرزندان گره تصادفی با احتمال برابر انتخاب می‌شوند. یک درخت بازی با بازیکنان A, B و C را در نظر بگیرید. در اینجا باید بررسی کنید چگونه می‌توان یک درخت expectimax را هرس کرد.

فرض کنید فرزندان هر گره از چپ به راست پیمایش می‌شوند. برای هر یک از درخت‌های بازی زیر مقادیری را به برگ‌ها اختصاص دهید به گونه‌ای که گره پررنگ شده بتواند هرس شود (مهم نیست اگر گره‌های بیشتری هم هرس شوند)، یا اگر چنین تخصیصی ممکن نیست بنویسید غیرممکن. می‌توانید مقادیر اختصاص دهید که باعث شود یکی از اجداد گره پررنگ شده هرس شود. توجه داشته باشید که نباید در حالت برابری هرس انجام دهید. برگ‌ها باید مقادیر متناهی داشته باشند. (جواب هر فرد به این سوال می‌تواند متفاوت باشد) (15 نمره)



بخش عملی (40 نمره):

1- بازی Gomoku یک بازی با دو بازیکن است که روی یک صفحه 15×15 انجام می‌شود. در این بازی دو بازیکن به نوبت سنگ‌های سیاه و سفید خود را روی خانه‌های صفحه قرار می‌دهند. هدف هر بازیکن این است که بتواند پنج سنگ متوالی از رنگ خود را به صورت

افقی، عمودی یا مورب در یک ردیف قرار دهد. بازیکنی که زودتر به این هدف برسد، برنده بازی است.

- در این تمرین شما باید یک عامل هوش مصنوعی مبتنی بر الگوریتم‌های نظریه بازی برای این بازی پیاده‌سازی کنید. تا بتواند به صورت خودکار تصمیم‌گیری کند. کد اصلی بازی و ساختار کلاس GomokuAI به شما داده شده است و بسیاری از بخش‌های عمومی مانند:

- ساخت و نمایش صفحه بازی
- بررسی حرکت‌های مجاز
- بررسی برنده بودن بازی
- و توابع ارزیابی اولیه
- از قبل پیاده‌سازی شده‌اند.

بنابراین تمرکز شما تنها بر روی پیاده‌سازی الگوریتم‌های تصمیم‌گیری هوشمند خواهد بود.

- ساختار پروژه:

- فایل [play.py](#) که محیط بازی را طراحی کرده و نیاز به تغییرات ندارد.
- پوشه source که دارای فایل‌های [utils.py](#) و [gomoku.py](#) و [Al.py](#) است که شما باید فایل [Al.py](#) و توابع `random_move`، `monteCarloTreeSearch`، `minimax` و `alphaBetaPruning` را پیاده کنید
- پوشه gui که دارای فایل‌های [button.py](#) و [interface.py](#) است و شما از این پوشه چیزی را نباید تغییر بدهید.
- در نهایت پوشه assets که برای گرافیک بازی است و دارای عکس‌ها هست.

- گزارش نهایی

- در پایان لازم است علاوه بر گزارش پیاده‌سازی یک گزارش مختصر آماده کرده که شامل این موارد باشد:
- مقایسه عملکرد چهار روش Random, Monte Carlo, AlphaBeta و miniMax
- زمان میانگین تصمیم‌گیری هر الگوریتم که پیاده‌سازی شده است.
- تعداد حرکات انجام شده در هر بازی
- نتایج کلی بازی‌ها