بسمه تعالی

هوش مصنوعی

نیم‌سال دوم ۹۷ـ۹۸

مدرس: مهدیه سلیمانی باغشاه

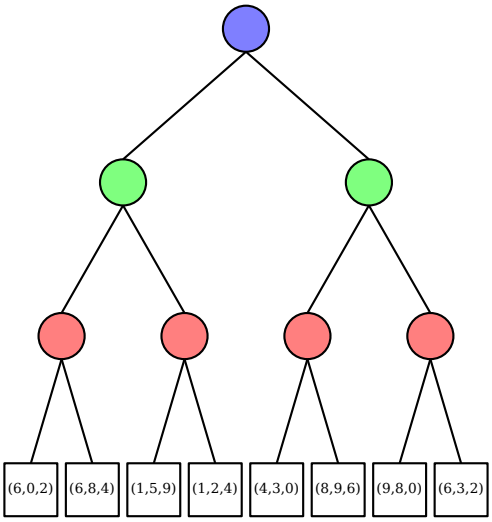
دانشکده مهندسی کامپیوتر

تمرین سوم

مهلت تحویل تمرین: ۹۸/۱/۱۷

تمرین‌های نظری ۲۵ نمره

۱. (۷ نمره) الگوریتم Minimax را می‌توان به بازی‌هایی که zero-sum نیستند و بیش از دو بازیکن دارند نیز تامیم داد. در درخت بازی زیر، سه بازیکن با رنگ‌های متفاوت نشان داده شده‌اند و هر کدام می‌خواهند امتیاز خود را بیشینه کنند. امتیاز مشخص‌شده در برگ‌ها به شکل (امتیاز بازیکن قرمز، امتیاز بازیکن سبز، امتیاز بازیکن آبی) می‌باشد. ارزش هر گره را با الگوریتم تامیم‌یافته‌ی Minimax به دست آورید. آیا در این حالت می‌توان از هرس آلفاـ بتا استفاده کرد؟



۲. (۸ نمره) درستی یا نادرستی موارد زیر را با ذکر دلیل بیان کنید:

(آ) در الگوریتم آلفاـ بتا ممکن است یکی از فرزندان ریشه هرس شود.

(ب) اگر بازی zero-sum نباشد و بازیکن دوم نابهینه بازی کند، بازیکن اول حتما می‌تواند امتیاز بیشتری کسب کند (نسبت به حالتی که بازیکن دوم بهینه بازی می‌کند).

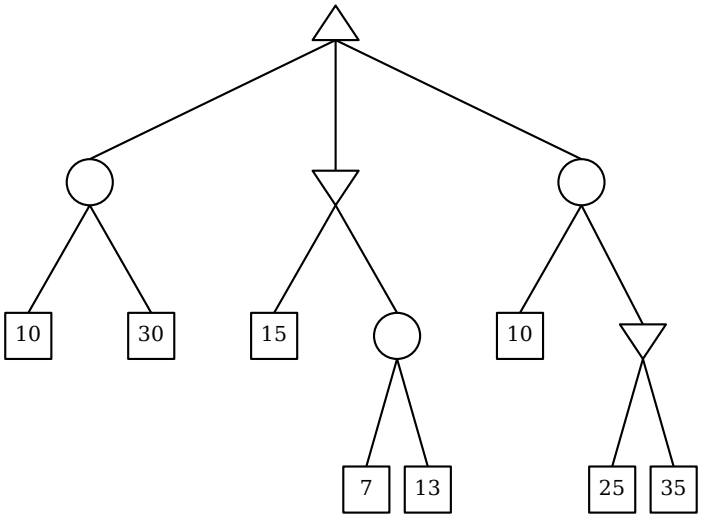
(ج) اگر MIN نابهینه بازی کند، MAX حتما می‌تواند امتیاز بیشتری کسب کند (نسبت به حالتی که MIN بهینه بازی می‌کند).

(د) اگر MIN تصادفی بازی کند، استراتژی بهینه‌ی MAX تغییر نمی‌کند (نسبت به حالتی که MIN بهینه بازی می‌کند).

۳. (۱۰ نمره) با توجه به شکل به سوالات زیر پاسخ دهید. فرض کنید احتمال رویدادها در گره‌های شانس برابرند.

(آ) (۳ نمره) ارزش هر گره را مشخص کنید.

(ب) (۷ نمره) الگوریتم آلفا ـ بتا کدام شاخه‌ها را هرس می‌کند؟ فرض کنید گره‌ها از چپ به راست بررسی می‌شوند.



تمرین عملی

۵۰ نمره

در این تمرین می‌خواهیم با استفاده از الگوریتم‌هایی که یاد گرفتیم یک حریف برای بازی Isolation پیاده‌سازی کنیم. این بازی به این صورت است که در یک صفحه‌ی ۷ × ۷، ۲ مهره‌ی اسب وجود دارد که می‌توانند به صورت L در صفحه حرکت کنند و در هر خانه‌ای از جدول که قرار بگیرند آن خانه را تخریب می‌کنند به صورتی که در حرکات بعدی خودشان یا حریفشان نمی‌توانند در آن خانه قرار بگیرند و عملا این خانه از جدول حذف می‌شود. در نهایت بازیکنی که در نوبت خودش نتواند هیچ حرکتی انجام دهد بازنده‌ی بازی است و بازیکن دیگر برنده‌ی این مسابقه است. شما باید برای این بازی یک عامل هوشمند طراحی کنید. برای طراحی این عامل، شما باید از درخت Minimax استفاده کرده و برای بهبود آن از هرس آلفا ـ بتا کمک بگیرید. برای پیاده‌سازی تابع Minimax لازم است که برای هر حالت بازی یک تابع ارزیابی داشته باشید. با توجه به اینکه بخشی از نمره‌ی شما به این قسمت اختصاص دارد و میزان هوشمندی برنامه‌ی شما را این قسمت مشخص می‌کند، سعی کنید که در طراحی آن دقت کنید. لازم به ذکر است که برنامه‌ی شما برای هر حرکت فقط ۱۵۰ میلی‌ثانیه زمان دارد. در نتیجه اگر انتخاب حرکت بیشتر طول بکشد، شما بازنده‌ی بازی خواهید بود. همچنین هر حرکت نامعتبر از جانب برنامه‌ی شما منجر به باخت شما خواهد شد.

توضیحات

- برای اطلاع کامل از نحوه‌ی اجرای بازی، فایل `README.md` و کدهای موجود را مطالعه کنید. برای درک بهتر کد، چند بازیکن و حریف تمرینی در قسمت `sample_players.py` قرار گرفته است.
- تنها چیزی که شما باید تحویل دهید، `game_agent.py` است. بنابراین مجاز به تغییر در سایر کدها نیستید.
- بخش عمده‌ای از نمره‌ی این تمرین به عملکرد کد شما در برابر کدهای سایر دانشجویان تعلق می‌گیرد.

موفق باشید