بازی Line'em Up یک نسخه عمومی از tic-tac-toe است: این یک بازی 2 نفره adeverserial است که روی n×n بازی می شود.

تابلویی که در آن هر موقعیت دارای یکی از 4 مقدار است:

• یک قطعه سفید (◦) – برای بازیکنی که مهره های سفید را می نوازد

• یک قطعه سیاه (•) – برای بازیکنی که مهره های سیاه را می نوازد

• یک بلوک () – جایی که هیچ بازیکنی نمی تواند مهره های خود را قرار دهد (در تیک تاک، هیچ بلوکی وجود ندارد)

• یک موقعیت خالی () – که بازیکنان می توانند مهره های خود را در آن قرار دهند

در موقعیت‌های خاصی قرار می‌گیرند و بقیه موقعیت‌ها خالی می‌مانند.

**تنظیم بازی**: در ابتدا، بلوک‌ها () در موقعیت‌های خاصی قرار می‌گیرند و بقیه موقعیت‌ها خالی می‌مانند.

هر بازیکن رنگی را که بازی می کند (◦ یا •) انتخاب می کند. بازیکنی که بازی می کند ◦ همیشه اول بازی می کند. شکل 1 نشان می دهد

صفحه اولیه بازی با n=5 و b=6 بلوک در [(0,D), (1,B), (2,A), (2,D), (3,B), (4,C) قرار دارند. )].

Table

Description automatically generated

شکل 1: نمونه ای از تخته اولیه با n=5 و b=6 بلوک در موقعیت های :

[(D,0),(1,B),(2,A),(3,B),(4,C)]

**حرکات**: بازیکنان در نوبت های متناوب بازی می کنند. در هر نوبت، بازیکن فعال یک موقعیت خالی را انتخاب می کند

آنها رنگ خود را قرار می دهند (◦ یا •). سپس بازیکن بعدی حرکت خود را انجام می دهد.

**پایان بازی**: به محض اینکه یکی از بازیکنان تکه های متوالی از رنگ خود را پشت سر هم ردیف کند، بازی به پایان می رسد.

در یک ستون یا مورب؛ در این صورت، آن بازیکن برنده بازی است. اگر هیچ موقعیت خالی دیگری در مورد وجود نداشته باشد

تخته، سپس بازی با تساوی به پایان می رسد. به عنوان مثال، با صفحه بازی شکل 1، بردهای احتمالی در آن نشان داده شده است

شکل های 2 و 3 در صورتی که s=4 باشد و یک رسم احتمالی در شکل 4 نشان داده شده است.

* Calendar

  Description automatically generated

**2-وظیفه شما:**

یک adverserial search برای بازی Line'em up برای مقادیر n، s و b داده شده به عنوان ورودی اجرا کنید.

2.1 **کد اسکلت(Skeleton code):**

برای شروع، یک کد اسکلتی که یک بازی استاندارد tic-tac-toe را پیاده سازی می کند (یعنی با n=3، b=0 و s=3)

این اسکلت شامل:

1. یک تابع مینی حداکثر(Min-Max function) کد از قبل،

2. یک تابع آلفا-بتا از قبل کدگذاری شده و

3. یک basic interface

کد حالت‌های برگ درخت بازی (Game-tree)را ارزیابی می‌کند، بنابراین برای مقادیر n>3 در واقعاً قابل استفاده نیست.

یک زمان معقول شما باید این کد را گسترش دهید:

1. ویژگی بازی Line'em Up شرح داده شده در بخش 1 را در نظر بگیرید،

2. توسعه 2 اکتشافی e1 و e2 - به بخش 2.3 مراجعه کنید،

3. تعدادی از پارامترها را به عنوان ورودی بپذیرید - بخش 2.4 و را ببینید

4. خروجی انواع اطلاعات - به بخش 4.1 مراجعه کنید.

**2.2 محیط برنامه نویسی**

برای برنامه ریزی پروژه باید از پایتون 3.7 استفاده کنید. با استفاده از PyPy1

زمان اجرا به جای CPython معمولی خواهد بود

کد خود را به میزان قابل توجهی سریعتر اجرا کنید، اما مراقب باشید که Pypy با همه ماژول های پایتون سازگار نیست

2.3 اکتشافی(The Heuristics)

دواکتشافی مختلف (e1 و e2) را برای این بازی توسعه دهید. توصیه می شود e1 بسیار ساده اما سریع باشدو e2 برای محاسبه پیچیده تر و پیچیده تر باشد. به این ترتیب، شمامی توانید درک بهتری از تقابل زمان اجرا و کیفیت تخمین به منظور برنده شدن در بازی با توجه به محدودیت های زمانی داشته باشید

**ورودی Input**

نیازی به داشتن یک user-interface فانتزی نیست. یک رابط command-line interface ساده با یک I/O مبتنی بر متن کافی است تا زمانی که برنامه شما از ورودی زیر پشتیبانی کند:

**پارامترهای بازی**: برنامه شما باید پارامترهای بازی زیر را بپذیرد.

(الف) اندازه تابلو { n { یک عدد صحیح در [3..10]

(ب) تعداد بلوک ها { b { یک عدد صحیح در [0..2n]

(ج) موقعیت های بلوک ها { b مختصات هیئت تابلو

(د) اندازه ترکیب برنده { s { یک عدد صحیح در [3..n]

(ه) حداکثر عمق جستجوی َAdverserial برای بازیکن 1 و برای بازیکن 2 { 2 اعداد صحیح d1 و d2

(f) حداکثر زمان مجاز (بر حسب ثانیه) برای برنامه شما برای بازگشت یک حرکت { t

هوش مصنوعی شما نباید بیش از t ثانیه طول بکشد تا حرکت خود را برگرداند. اگر این کار را کرد، هوش مصنوعی شما به طور خودکار بازنده می شود.این مستلزم آن است که حتی اگر جستجوی Adverserial شما اجازه داشته باشد تا عمق d در

درخت بازی برود، ممکن است هر بار برای انجام این کار وقت نداشته باشد. برنامه شما باید زمان را کنترل کند و اگرزمان کافی برای کاوش در تمام حالات در عمق d باقی نمانده است، باید جستجوی خود را در عمق d قطع کند و مقادیر را برای حالت های باقی مانده به سرعت قبل از اتمام زمان برمی گرداند.

(ز) یک بولی برای اجبار استفاده از minmax(false) یاalphabeta(true)

(ح) حالت های بازی

برنامه شما باید بتواند هر یک از بازیکنان را انسان یا هوش مصنوعی کند. این به این معنی است که شما باید

بتوانید برنامه خود را در هر 4 ترکیب پلیر اجرا کنید: H-H، H-AI، AI-H و AI-AI

.)انسان- انسان)و (انسان-AI), (AI-انسان)و

((AI-AI

**مختصات حرکت ها:** اگر حداقل یک بازیکن یک انسان باشد، حرکات آنها توسط اولین بار مشخص می شود.

نشان دهنده ستون (با شماره [A..J])، سپس ردیف (با شماره [0..(n-1)])، برای مثال B 3. توجه داشته باشید

که اگر یک بازیکن انسانی وارد یک حرکت غیرقانونی شود (مثلا W 3 یا مختصات موقعیتی که خالی نیست)،

سپس فقط به آنها اخطار داده می شود و به آنها فرصتی داده می شود تا بدون جریمه وارد حرکت دیگری شوند. با این حال، اگر

برنامه شما یک حرکت غیرقانونی ایجاد می کند، سپس به طور خودکار بازی را از دست می دهد.

2.**5 خروجی**

برنامه شما باید 2 نوع خروجی تولید کند: ردیابی بازی و یک تابلوی امتیاز با آمار چندگانه

بازی ها. نمونه های خروجی به پیوست ارسال میگردد. اکنون آنها را دانلود کنید تا بهتر بتوانید توضیحات زیر را دنبال کنید

2.5.1 فایل های ردیابی بازی

ردیابی بازی برای نشان دادن تکامل یک بازی است که در آن هر دو بازیکن توسط یک هوش مصنوعی کنترل می شوند. این

نام این خروجی‌ها باید از قالب پیروی کند: gameTrace-<n><b><s><t>.txt (برای مثال gameTrace-5642.txt).

هر اثر باید شامل موارد زیر باشد:

1. پارامترهای بازی: مقادیر n، b، s، t

2. موقعیت بلوک ها

3. پارامترهای هر بازیکن: مقادیر انسان یا AI و در مورد هوش مصنوعی مقادیر مربوطه

از عمق جستجو (d1 یا d2)، مقادیر مربوط به حروف الفبا در مقابل حداقل جستجو

(Alphabeta vs minimax)

(a1 یا a2 یا هر دو) و کدام اکتشافی استفاده شده است (e1 یا e2).

4. نمایش تنظیمات اولیه تابلو.

5. سپس، برای هر حرکت، نمایش داده شود:

(الف) حرکت انجام شده (به عنوان مثال B 4)

(ب) تشکیلات جدید تابلو

ج) آمار زیر:

-زمان ارزیابی اکتشافی (بر حسب ثانیه)

- تعداد حالت های ارزیابی شده توسط تابع اکتشافی

کد شما باید تعداد حالت‌هایی را که تابع اکتشافی شما برای انتخاب حرکت فعلی خود ارزیابی کرده است شمرده و برگرداند(تحویل دهد)

-تعداد حالت های ارزیابی شده در هر عمق (ریشه را در عمق 0 در نظر بگیرید)

-تعداد حالت های ارزیابی شده در هر عمق. در اینجا شما نشان می دهید که چه تعداد گره ارزیابی شده است در عمق 1، در عمق 2، . . .

-عمق متوسط ​​(AD) ارزیابی اکتشافی در درخت

همانطور که در بالا توضیح داده شد، برنامه شما باید در عرض t ثانیه در مورد حرکت بعدی تصمیم بگیرد. از این رو جستجوتمام گره ها در سطح d ممکن است امکان پذیر نباشد. در اینجا، شما باید میانگین عمق همه را محاسبه کنید

-حالت های ارزیابی شده توسط تابع اکتشافی شما.

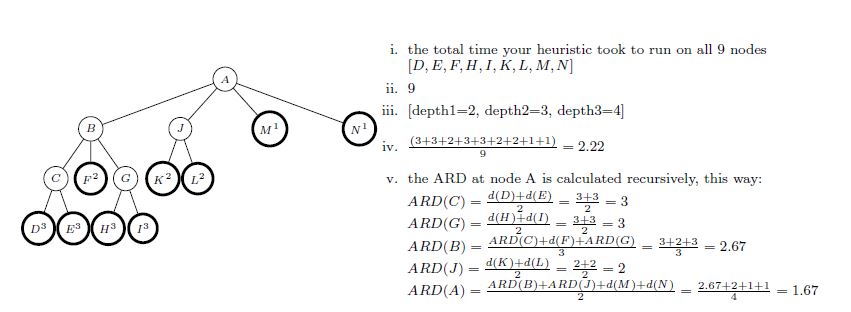
- میانگین عمق بازگشت (ARD) در وضعیت فعلی

برای توضیح به مثال زیر مراجعه کنید.

مثال زیر این آمار را نشان می دهد. فرض کنید جستجوی شما اکتشافی را اجرا کرده است

روی گره ها به صورت پررنگ درخت بازی زیر عمل کنید. عمق این گره ها به صورت a نشان داده شده است

بالانویس در کنار نام گره. آماری که باید نمایش داده شود عبارتند از:



سپس در پایان بازی، موارد زیر را نمایش دهید:

(الف) برنده

(ب) برای هر اکتشافی، آمار زیر:

من. میانگین زمان ارزیابی اکتشافی برای هر حالت ارزیابی شده (بر حسب ثانیه)

دوم تعداد حالت های ارزیابی شده توسط تابع اکتشافی در طول کل بازی

سوم میانگین عمق میانگین هر حرکت ارزیابی اکتشافی در درخت {یعنی میانگین

از آمار 4(c)iii در بالا برای همه حرکات

iv تعداد کل حالت های ارزیابی شده در هر عمق در طول کل بازی

v. میانگین عمق بازگشت متوسط ​​در هر حرکت - یعنی میانگین آماره 4(c)v در بالا برای

همه حرکات

vi. تعداد کل حرکات در بازی

**2.5.2 فایل جدول امتیاز**

شما یک سری بازی rX2 (به عنوان مثال 2X 10)را در حالت هوش مصنوعی در مقابل هوش مصنوعی اجرا خواهید کرد که در آن هر بازیکن از یک بازی متفاوت استفاده می کند.

اکتشافی (به عنوان مثال، بازیکن 1 از e1 استفاده می کند و بازیکن 2 از e2 استفاده می کند)، و هر بازیکن به صورت r بار بازی می کند، و به عنوان زمان r بازی می کند.

شما اطلاعات زیر را در خروجی ای به نام scoreboard.txt ذخیره خواهید کرد:

1. پارامترهای بازی (مقادیر n، b، l، t)

2. پارامترهای هر بازیکن: حداکثر عمق جستجوی آنها (d1 و d2))، حروف الفبا(Alphabeta) یا حداقل آنها(minimax) (a1 و a1) و اکتشافی آنها

3. تعداد بازی های انجام شده (مقدار 2X r)

4. تعداد و درصد بردهای اکتشافی e1 و اکتشافی e2

5. تمام اطلاعات نمایش داده شده در پایان یک بازی (به بخش 2.5.1 مراجعه کنید)، اما به طور میانگین در بازی های 2 ثانیه است.

برای هر سری از بازی‌ها، آمار سری را در انتهای scoreboard.txt le اضافه کنید.

هنگامی که کد شما اجرا شد، یک نمونه برای یک رد بازی برای تنظیمات زیر ایجاد کنید:

1. gameTrace-4435.txt (یعنی: n=4، b=4، s=3، t=5)

با d1=6، d2=6، a1=FALSE، a2=FALSE با بلوک های [(0.0)، (0.4)، (4.0)، (4،4)]

2. gameTrace-4431.txt: با d1=6، d2=6، a1=TRUE، a2=TRUE با بلوک های [(0.0)، (0.4)، (4.0)، (4، 4)]

3. gameTrace-5441.txt: با d1=2، d2=6، a1=TRUE، a2=TRUE با بلوک ها در موقعیت های تصادفی

4. gameTrace-5445.txt: با d1=6، d2=6، a1=TRUE، a2=TRUE با بلوک ها در موقعیت های تصادفی

5. gameTrace-8551.txt: با d1=2، d2=6، a1=TRUE، a2=TRUE با بلوک ها در موقعیت های تصادفی

6. gameTrace-8555.txt: با d1=2، d2=6، a1=TRUE، a2=TRUE با بلوک ها در موقعیت های تصادفی

7. gameTrace-8651.txt: با d1=6، d2=6، a1=TRUE، a2=TRUE با بلوک ها در موقعیت های تصادفی

8. gameTrace-8655.txt: با d1=6، d2=6، a1=TRUE، a2=TRUE با بلوک ها در موقعیت های تصادفی

و یک scoreboard.txt برای یک سری 10 بازی با استفاده از تنظیمات مشابه بالا (یعنی 10 بازی با

conguration (1) + 10 بازی با conguration (2) . . . )

تابلوی امتیاز خود را برای مقایسه و مقایسه اثر عمق، آلفا-بتا و دو اکتشافی تجزیه و تحلیل کنید.

چند اسلاید برای توضیح اکتشافات خود و ارائه و تجزیه و تحلیل داده های فوق آماده کنید. ارائه خواهید کرد

این اسلایدها در نسخه نمایشی (به بخش 4.2 مراجعه کنید).

**ارسال**

تمام فایل های لازم برای اجرای کد خود را علاوه بر یک readme.md که حاوی دستورالعمل های مشخص و کامل در مورد نحوه اجرای آزمایش های خود است ارسال کنید. اگر دستورالعمل های موجود در فایل readme شما کار نکند، ناقص باشند

یا فایلی گم شده باشد، منفعت شک به شما داده نمی شود. 8 رد بازی و جدول امتیازات را ایجاد کنید

فایل های خروجی مشخص شده در بخش 2.6.