

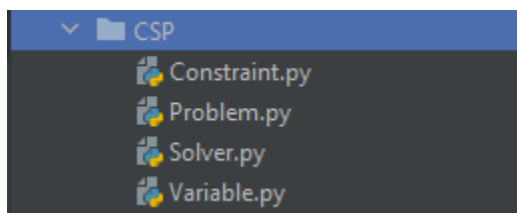
پروژه ارضای محدودیت درس مبانی هوش مصنوعی | بهار ۱۴۰۴

توضیحات پروژه

در این پروژه، شما باید یک حل کننده مسئله‌های ارضای محدودیت را به طور عام‌منظوره پیاده سازی کنید. همراه با توضیحات پروژه، کدهای اولیه پروژه در اختیار شما قرار خواهد گرفت. در این کدها ساختار و کلاس بندی برنامه مورد نظر به همراه دو مسئله CSP نمونه پیاده سازی شده است. ساختار پروژه در ادامه و درس جلسه توضیح پروژه تشریح خواهد شد. خواسته این پروژه از شما پیاده‌سازی کلاس Solver برای حل مسائل ارضای محدودیت است.

ساختار پروژه

در شکل زیر فایل‌ها و کلاس‌های مربوطه به این پروژه را مشاهده می‌کنید.



همانطور که می‌دانید، در هر مسئله ارضای محدودیت، تعدادی متغیر و تعدادی محدودیت وجود دارد. که کلاس‌های متناظر آنها ساخته شده است. هر متغیر یک نام و یک دامنه دارد که به شکل لیستی از مقادیر قابل انتصاب به متغیر نشان داده می‌شود. کلاس محدودیت، یک کلاس آبسترکت یا انتزاعی است که لازم است برای هر یک از مسائل شخصی سازی شود اما به طور کلی شامل تعداد متغیر و تابعی به نام `is_satisfied` می‌باشد که مشخص می‌کند که آیا محدودیت ارضا شده است یا نه. همچنین کلاس Solver در پروژه وجود دارد که شما باید بدنه آن و متد `Solve` را در آن پیاده سازی کنید.

مسائل پیاده سازی شده

حل کننده شما باید قابلیت حل هر نوع مسئله ارضای محدودیت را داشته باشد و لذا در تحویل پروژه مسائلی غیر از مسائلی که به طور نمونه پیاده سازی شده و در اختیار شما قرار داده می‌شود، مسائلی دیگری به برنامه شما داده می‌شود و برنامه شما باید بتواند بدون مشکل مسئله را حل کند. مسائلی که به عنوان نمونه پیاده سازی شده‌اند به شرح زیر است:

مسئله رنگ آمیزی نقشه



در مسئله رنگ آمیزی نقشه که احتمالاً از کلاس درس با آن آشنا هستید، هدف رنگ آمیزی نقشه استان های استرالیا با سه رنگ آبی، قرمز و سبز است به نحوی که هیچ دو استان مجاور هم رنگ نباشند. این مسئله برای شما پیاده سازی شده است.

مسئله سانتای مخفی (Secret Santa)

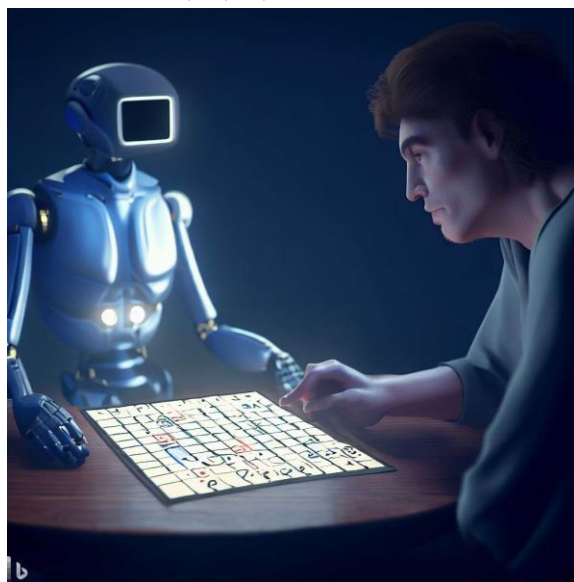


Secret Santa یک رسم در عید کریسمس است که در طی آن، اعضای گروهی از دوستان، خانواده یا همکاران، از یک کیسه شامل نام های افراد شرکت کننده، یک نام بیرون آورده و سپس باید به شخصی که نامش خارج شده است در روز عید کریسمس هدیه بدهند. واضح است که هدیه گیرنده نمی تواند برای دو نفر مشترک باشد. هدیه گیرنده پس از باز

کردن هدیه خود، باید حدس بزند که کدام یک از اعضای گروه Secret Santa آنها بوده است. این یک رسم کلاسیک است که در آن حدس زدن هدیه دهنده به اندازه دریافت هدیه بخشی از سرگرمی است.

در نمونه کد اولیه که در اختیار شما قرار می‌گیرد، این مسئله مدل سازی شده است. به این شکل کلاس مربوط به این مسئله یک لیست از اسامی افراد شرکت‌کننده در این بازی دریافت کرده و محدودیت های مورد نظر را ایجاد می‌کند. حل کننده شما باید قادر باشد، این مسئله را با هر تعداد اسم با سرعت خوبی حل کند.

مسئله سودوکو



سودوکو یک بازی پازل محبوب است که شامل پر کردن یک شبکه 9×9 با اعداد از ۱ تا ۹ است. این شبکه به ۹ شبکه فرعی 3×3 به نام "جعبه" تقسیم می‌شود و هر سطر و ستون شبکه باید شامل تمام اعداد از ۱ تا ۹ بدون تکرار هدف از این بازی این است که بر اساس مجموعه ای از سرخ های داده شده، شبکه را با اعداد صحیح پر کنید.

مسئله سودوکو را می‌توان به عنوان یک مسئله رضایت محدودیت (CSP) فرموله کرد، که چارچوبی برای نمایش و حل مسائل است که در آن هدف یافتن مجموعه ای از تکالیف متغیر است که مجموعه ای از محدودیت ها را برآورده می‌کند. در مورد سودوکو، متغیرها سلول های شبکه هستند و محدودیت ها عبارتند از:

هر خانه باید دارای یک عدد از ۱ تا ۹ باشد.

هر ردیف باید شامل تمام اعداد از ۱ تا ۹ بدون هیچ تکراری باشد.

هر ستون باید شامل تمام اعداد از ۱ تا ۹ بدون هیچ تکراری باشد.

هر کادر باید شامل تمام اعداد از ۱ تا ۹ بدون تکرار باشد.

مسئله معجون‌های جادویی (نمره اضافه و اختیاری)



پادشاه سعی دارد از ۵ معجون تهیه شده توسط ۵ شیمیدان به نام های آلدو، بئاتریس، ایگناتیوس، لورنزو و اورسولا یک معجون شفابخش بسازد. این معجون ها به رنگ های سبز، آبی، قرمز، سیاه و بنفش هستند و هر کدام اثر منحصر به فردی دارند: اسید، سم، شفابخش، دگرگونی و ناهرئی. پادشاه باید تعیین کند که کدام شیمیدان هر معجون را تهیه کرده و هر معجون چه تأثیری دارد.

با این حال، پادشاه اطلاعات اضافی دارد که مشکل را چالش برانگیزتر می کند:

۱. معجون آلدو قرمز و سبز نیست، بلکه معجون دگرگونی است.
۲. معجون بئاتریس آبی است و معجون اسیدی یا شفابخش نیست.
۳. معجون ایگناتیوس بنفش یا سیاه نیست، بلکه معجون سمی است.
۴. معجون لورنزو سبز است و معجون سمی نیست.

۵. معجون اورسولا سیاه یا آبی نیست، بلکه معجون نامرئی است.

پیشنهاد می‌شود علاوه بر مسائل پیاده سازی شده، مسائلی دیگری را تعریف و با استفاده از حل‌کننده خود حل کنید تا از قابلیت‌های برنامه خود مطمئن شوید.

الگوریتم‌هایی که باید پیاده سازی شوند

همانطور که در بالاتر اشاره شد، شما باید بدنه کلاس Solver را تکمیل کرده و با استفاده از الگوریتم‌ها و ابتکارهای زیر مسائل را حل کنید:

۱. Backtracking

۲. Forward Checking

۳. MRV

۴. LCV

همچنین لازم است که زمان اجرای حل را اندازه گیری کرده و در پایان حل نمایش دهید.

پیشنهاد می‌شود که الگوریتم‌های و توابع ابتکاری را به ترتیبی که در بالا آمده است پیاده سازی کنید. لازم است که حل‌کننده شما به گونه‌ای پیاده سازی شود که امکان فعال یا غیرفعال کردن قابلیت‌های ۲ تا ۴ را داشته باشد تا بشود تاثیر آنها را در زمان حل بررسی کرد و نمایش داد. (عملیات مقایسه زمان اجرا در تحویل پروژه از شما خواسته خواهد شد)

نکات تکمیلی

- پیاده سازی مسئله معجون‌های جادویی اختیاری و دارای نمره اضافه می‌باشد. (۱۵ درصد)
- استفاده تدریجی از Git در روند پیاده سازی پروژه دارای نمره اضافه خواهد بود. (۱۰ درصد)