



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر

استاد درس: دکتر مهدی قطعی

استاد کارگاه: بهنام یوسفی مهر

فروردین ۱۴۰۳

راهنمای تمرین ۴

درس هوش مصنوعی و کارگاه



توضیح تمرین

در اسلاید های درس با مساله رنگ امیزی نقشه آشنا شده اید. در این مساله هدف داریم تا نقشه مسطح یک کشور را به گونه ای رنگ امیزی کنیم تا هیچ دو شهر مجاوری رنگ یکسان نداشته باشند. در واقع میخواهیم یک گراف مسطح را به گونه ای رنگ کنیم که هیچ دو راس مجاور هم رنگ نباشند.

همانطور که میدانید یک گراف مسطح را میتوان با ۴ رنگ به گونه ای رنگ کرد که هیچ دو راس مجاوری هم رنگ نباشند، به همین علت به این مساله، مساله ۴ رنگ نیز گفته میشود.

این مساله به طور کامل در اینجا بیان شده است.

حال در این پروژه میخواهیم این مساله را به صورت یک مساله csp نگاه کرده و با استفاده از الگوریتم های حل csp این مساله را حل کنیم. مساله ۴ رنگ به صورت csp در اینجا بیان شده است.

در این پروژه هدف داریم تا نقشه هریک از قاره های جهان را به گونه ای رنگ امیزی کنیم که هیچ دو کشور مجاوری رنگ یکسان نداشته باشند.

برای پیاده سازی این مساله ابتدا به صورت ساده با الگوریتم پس گرد این مساله را پیاده سازی کرده و سپس به بهبود الگوریتم میپردازیم.

بخش اول

در این بخش انتظار می رود تا با استفاده از روش پس گرد به حل این مساله بپردازید. برای مطالعه این روش میتوانید به این سایت مراجعه کنید.

پیاده سازی این بخش باید در قسمت مشخص شده در فایل کدی که در اختیارتان قرار گرفته است انجام شود (در ادامه به توضیح تمپلیت نیز میپردازیم).

بخش دوم

انتظار می رود در این قسمت روش پس گرد را با الگوریتم های CSP مطرح شده در کلاس بهبود داده و در هر بهبود نتیجه را ذخیره کرده و آن ها را با هم مقایسه کنید. در این بخش باید از هیوریستیک های MRV و یا LCV برای بهبود و افزایش سرعت بیشتر استفاده کنید همچنین AC۳ را نیز پیاده سازی کنید. هریک از این بخش ها در فایل تمپلیت محل پیاده سازی مربوط به خود را دارند.



مساله رنگ آمیزی پیشرفته

پس از پیاده سازی مساله رنگ آمیزی به صورت کلاسیک می‌خواهیم این مساله را کمی پیشرفته تر کنیم. همانطور که پیش تر بیان شد در مساله رنگ آمیزی می‌خواهیم به گونه ای رنگ آمیزی کنیم تا هیچ دو راس مجاور رنگ یکسان نداشته باشند. حال می‌خواهیم این شرط را به همسایه همسایه نیز تعمیم دهیم، در این بخش متغیر Neighborhood-distance را تعریف می‌کنیم که این متغیر نشان می‌دهد تا چند درجه همسایگی نباید راس ها هم‌رنگ باشند. در واقع یک نود را در نظر بگیرید همسایه های این راس همسایه های درجه یک آن هستند، همسایه همسایه های این راس همسایه های درجه دو این راس هستند و به همین صورت برای همسایه درجه ۳. حال متغیر Neighborhood-distance نشان می‌دهد تا چند درجه همسایگی نباید راس ها هم‌رنگ باشند.

در مساله ما که رنگ آمیزی نقشه می‌باشد مساله به این صورت می‌شود: ایران با افغانستان همسایه می باشد و افغانستان با چین همسایه است پس چین همسایه درجه دو ایران است (همسایه همسایه) و ایران با چین نباید رنگ یکسانی داشته باشند. در این مرحله از پروژه می‌خواهیم به پیاده سازی الگوریتم ها تا به اینجا این بخش را هم اضافه کنیم. از آنجایی که این هدف جدید برای رنگ آمیزی باعث افزایش پیچیدگی مساله می‌شود پس برای اجرا این بخش از توابع هیوریستیک و بهینه ساز که در قبل پیاده سازی کرده اید استفاده کنید.

توضیحات فایل تمپلیت

برای راحتی پیاده سازی و همچنین مقایسه راحت تر الگوریتم های پیاده سازی شده یک تمپلیت برای پروژه در اختیار شما قرار گرفته است که شامل بخش هایی است که در ادامه به طور مختصر هریک را توضیح می‌دهیم. این تمپلیت شامل گرافیک نیز می باشد تا برای هر نقشه قاره ای که ورودی می‌دهید، رنگ آمیزی براساس الگوریتمی که نوشته اید را به همراه تعداد assignment ها مشاهده کنید. توضیحات هربخش از کد و توابع موجود به طور کامل در داخل کد آمده است و همچنین فایل readme کد نیز به توضیح مختصر هرتابع پرداخته است.

• csp.py

این بخش یک object از مساله csp می‌سازد که این object شامل قیود و متغیر های مساله می باشد. شما باید توابع موجود در این کلاس را براساس توضیحات موجود در کد پیاده سازی کنید.

• graphics.py

این بخش برای نمایش نقشه رنگ آمیزی شده می باشد. نقشه رنگ شده یک قاره را ورودی گرفته و آن را رسم می‌کند.



نقشه ی همه ی قاره ها و گرافیک پیاده سازی انها موجود است و نیازی به تغییر این بخش ها در کد نیست و فقط کافی است پس از پیاده سازی الگوریتم ها، در ورودی اسم قاره ای که مدنظرتان است را بدهید تا نقشه را با الگوریتمی که پیاده سازی کرده اید رنگ آمیزی کند.

• map-generator.py

این بخش شامل تابعی است که یک دیکشنری از همسایگی ها میسازد. یعنی به هرکشور همه ی همسایه هایش را نسبت میدهد.

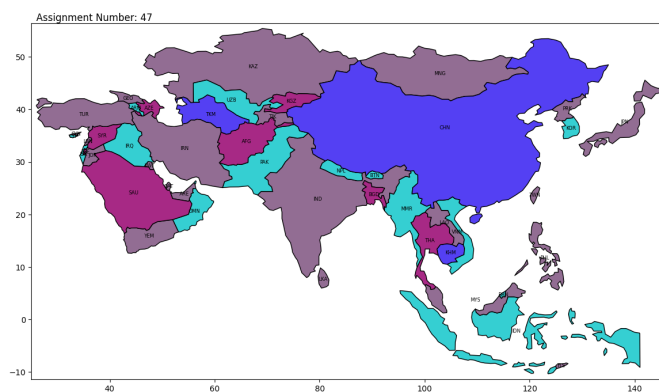
• Solver.py

این بخش شامل توابعی است که شما باید ان ها را پیاده سازی کنید. هریک از توابع backtrack-solver و arc-reduce و applyAC3 و MRV و LCV را در این بخش پیاده سازی کنید.

• main.py

این بخش main فایل می باشد که در این بخش با استفاده از فراخوانی مناسب توابع براساس ورودی ها باید نقشه نمایش داده شود. برای اجرا کد باید این بخش را نیز تکمیل کنید و از توابع موجود برای اجرا الگوریتم و رسم نقشه استفاده کنید. نحوه اجرا کد و ورودی دادن برای رنگ آمیزی و مشخص کردن الگوریتم ها به طور کامل در بخش readme در فایل کد تمپلیت توضیح داده شده است.

نمونه خروجی

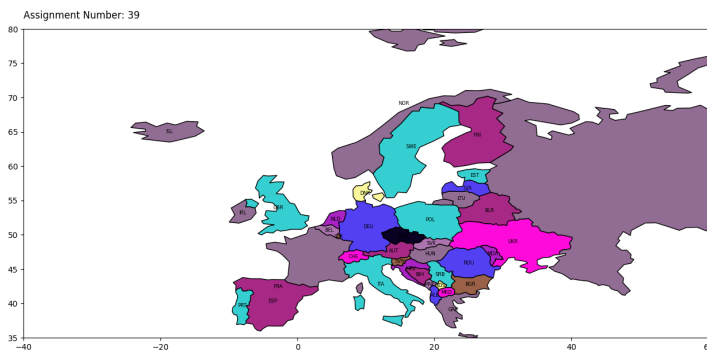
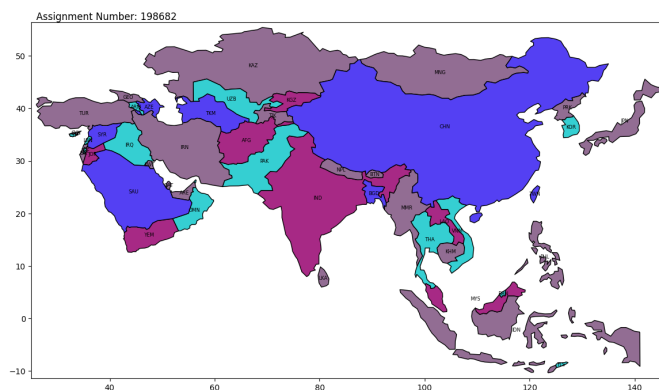




شکل بالا نمونه خروجی نقشه آسیا براساس یک درجه همسایگی می باشد یعنی کشور هایی که باهم همسایه هستند هم رنگ نمی باشند.

همانطور که در شکل میبینید در خروجی تعداد assignment ها نیز نمایش داده شده است که برابر است با تعداد دفعاتی که الگوریتم به متغیر ها مقدار داده است.

برای مثال در این مساله هر بار که در الگوریتم رنگ امیزی، رنگی به یک کشور داده میشود، به این متغیر یکی اضافه میشود. با استفاده از این مقدار میتوانید میزان بهبودی که روش های هیوریستیک و بهینه ساز در الگوریتم پسگرد دارند را مشاهده کنید. برای مثال شکل زیر رنگ امیزی همین نقشه با استفاده از روش پسگرد به تنهایی می باشد.



شکل بالا نمونه رنگ آمیزی قاره اروپا با درجه همسایگی دو می باشد. یعنی در این شکل همسایه همسایه ها نیز هم رنگ نمی باشند.



معیار های ارزیابی

- پیاده سازی کلاس های CSP و Solver
 - پیاده سازی تمام توابع گفته شده در کلاس csp
 - پیاده سازی تابع backtrack در کلاس Solver
- پیاده سازی توابع اکتشافی دامنه و متغیر
- پیاده سازی تابع های مربوط به الگوریتم ac3
- بررسی، تحلیل و مقایسه الگوریتم ها پیاده سازی شده با پارامتر های مختلف برای قاره (آسیا، اروپا، آمریکا، آفریقا) و درجه همسایگی (مقادیر ۱، ۲، ۳) و چاپ کمترین تعداد رنگی که برای رنگ آمیزی استفاده شده است (این مقدار برای مساله اصلی همواره ۴ است، اما برای رنگ آمیزی پیشرفته و در نظر گرفتن درجه همسایگی مقدار بزرگتر از ۴ است و باید برای حالت های مختلف تعداد رنگ های استفاده شده را خروجی دهید)، نحوه رنگ آمیزی، تعداد assignment ها و زمان مصرفی برای یافتن جواب را بررسی کنید.
- توضیحات کامل در قالب پایان نامه (شامل توضیح الگوریتم ها، توابع و نتایج تحلیل و مقایسه)
- علاوه بر بخش عملی، پاسخ دادن ب سوالات تیوری این تمرین نیز بخشی از ارزیابی خواهد بود و در ارایه مجازی تمرین چهارم باید به سوالات تیوری نیز تسلط داشته باشید.
- استفاده از مدل های زبانی برای پاسخگویی به تمرین غیرمجاز می باشد.

مهلت تحویل

- مهلت تحویل تمرین تا ساعت ۲۳:۵۹ روز ۳۱ فروردین می باشد.
- با استفاده از این **لینک** میتوانید به درایو پروژه چهارم دسترسی داشته باشید. سوالات تیوری و تمپلیت کد در این درایو می باشند.
- لطفا فایل pdf گزارش کار و فایل پیاده سازی کد و pdf پاسخ به سوالات تیوری را در قالب یک فایل zip در سامانه کورسز و کویرا بارگذاری نمایید.