

به نام خدا

پروژه شماره ۲ درس هوش مصنوعی - رنگ آمیزی نقشه در قالب حل مسئله ارضای محدودیت (CSP)

استاد: جناب آقای دکتر خدنگی

مقدمه: تعریف مسئله ارضای محدودیت

یک مسئله ارضای محدودیت به صورت یک $\langle X, D, C \rangle$ تعریف می گردد که:

مجموعه متغیر ها: $X = \{X_1, \dots, X_n\}$

مجموعه دامنه های مقادیر هر متغیر: $D = \{D_1, \dots, D_n\}$

مجموعه محدودیت ها: $C = \{C_1, \dots, C_m\}$

می باشد.

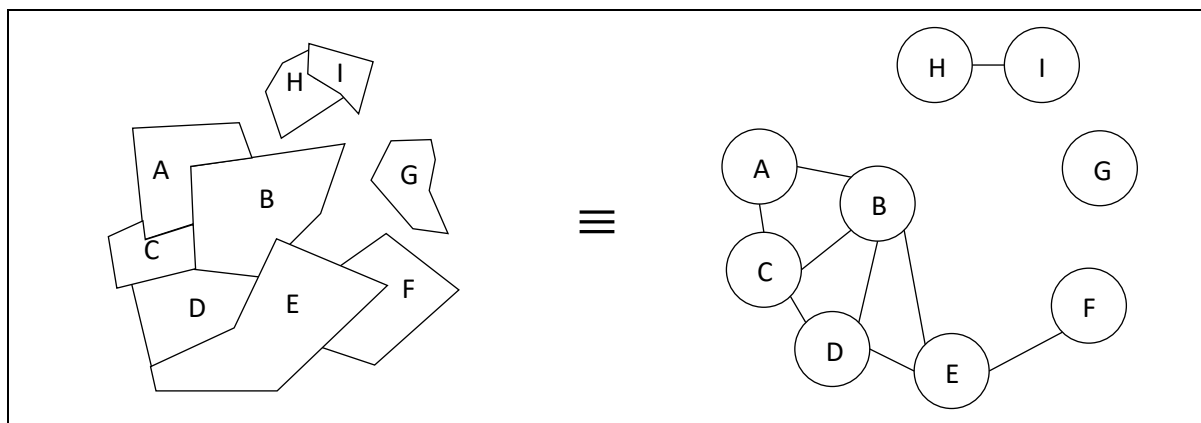
اهداف پروژه

۱. رنگ آمیزی نقشه در قالب حل مسئله CSP (که دارای محدودیت های باینری است) با استفاده از الگوریتم Backtracking

۲. مقایسه زمانی انواع بهبود بخشیدن ها (از طریق افزودن Filtering و Ordering) و نتیجه گیری از آن

شرح پروژه

می خواهیم در یک نقشه جغرافیایی فرضی، کشور های موجود در آن را رنگ آمیزی کنیم به نحوی که هیچ دو کشور همسایه ای دارای رنگ یکسان نباشند. نقشه را به صورت یک گراف، که نود های آن کشور ها بوده و یال های آن مشخص کننده همسایه بودن آن کشور ها است، داریم.



در این پروژه، محدودیت ها برای مسئله رنگ کردن نقشه به صورت باینری می باشد. یعنی مجموعه محدودیت ها به صورت:

$$X_i \neq X_j \quad \forall (i, j) \in NE$$

می باشد. NE مجموعه ای است که نشان می دهد کدام جفت متغیر ها با هم همسایه هستند. به بیان دیگر، هیچ دو کشوری در نقشه نبایستی دارای مقداری یکسان از دامنه خود باشند.

قوانین پروژه

- ✓ مسئله با استفاده از الگوریتم جستجوی Backtracking (بر مبنای الگوریتم DFS) حل شود.
- ✓ در Ordering پیاده سازی Forward Checking الزامی است.
- ✓ در Filtering پیاده سازی هیوریستیک Minimum Remaining Value الزامی است. (مربوط به ترتیب انتخاب متغیر ها)

(نمره امتیازی) در Ordering پیاده سازی Arc Consistency اختیاری و دارای نمره امتیازی است.

(نمره امتیازی) در Filtering پیاده سازی هیوریستیک Least Constraining Value اختیاری و دارای نمره امتیازی است. (مربوط به ترتیب انتخاب مقادیر از دامنه)

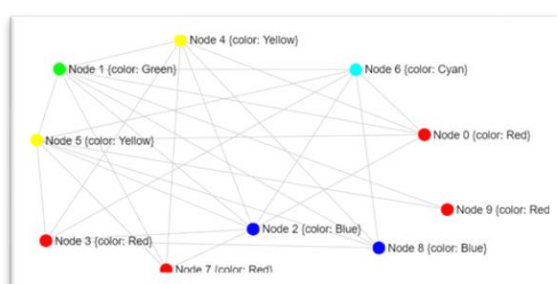
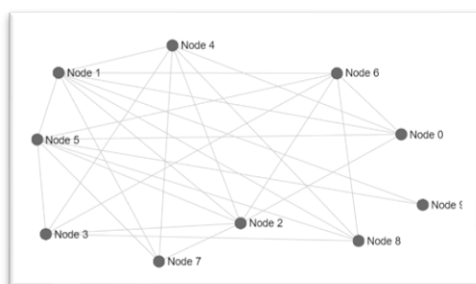
- ✓ یک گزارش (مثلا در قالب فایل ورد یا ویدیو)، میانگین های زمان اجرای ترکیب های مختلف را با هم برای یک نقشه خاص در یک نمودار ستونی مقایسه کنید. به بیان دیگر الگوریتم Backtracking را با افزودن ویژگی های جدول زیر مقایسه کنید. در انتها با استفاده از اطلاعات به دست آمده، در چند جمله به نتیجه گیری بپردازید. مانند پروژه قبلی، پیشنهاد می شود در کد میانگین ها را محاسبه کنید تا کارتان سرعت بیشتر و خطای محاسباتی کمتری نسبت به حالت دستی داشته باشد. هنگام تحویل پروژه، لزومی ندارد این قابلیت در اجرای برنامه باشد و صرفاً فایل گزارش همراه پروژه باشد.

Filtering	Ordering	
-	-	ستون اول نمودار
-	Forward checking	ستون دوم نمودار
Minimum Remaining Value	-	ستون سوم نمودار
Minimum Remaining Value	Forward checking	ستون چهارم نمودار

(نمره امتیازی) اگر هر گونه پیاده سازی امتیازی دیگری (مانند Arc Consistency) داشتید با ویژگی های بالا ترکیب کنید و در ستون های بعدی نمایش دهید.

نکات پیاده سازی

- ✓ پروژه به صورت غیر گرافیکی است. (مثلا در سی شارپ، پروژه Console Application کافی است)
- (نمره امتیازی) هر گونه پیاده سازی گرافیکی (ویندوزی یا تحت وب) برنامه دارای نمره امتیازی است.



- ✓ انتخاب زبان برنامه نویسی برای این پروژه آزاد است.

(نمره امتیازی) پیاده سازی در متلب به صورت گرافیکی یا غیر گرافیکی دارای نمره امتیازی است. منظور از پیاده سازی گرافیکی، میتواند plot کردن نتیجه باشد. اگرچه، پیاده سازی یک App توسط App Designer متلب دارای امتیاز بیشتری است.

✓ برنامه باید جزییات نقشه را یک فایل با نام **input.txt** بخواند و نتایج اجرا را در فایل با نام **output.txt** ذخیره کند.

✓ ساختار فایل **input.txt** به صورت زیر است که در زیر یک مثال از آن آورده شده است.

- **خط اول:** دامنه (برای مثال اسم رنگ ها) به صورت یک رشته تفکیک شده با ویرگول
- **خطوط بعدی:** ماتریس مجاورت گراف نقشه

رنگ های مجاز برای رنگ آمیزی

ماتریس مجاورت یک گراف با ۱۰ نود می باشد. در این جا، یک نقشه با ۱۰ کشور داریم که عدد ۱ نشان دهنده همسایه بودن دو کشور است و عدد ۰ نشان

```
input.txt
1 Red,Green,Blue,Yellow,Cyan,Purple,Black
2 0110111000
3 1010111111
4 1101111100
5 0010111010
6 1111000110
7 1111001111
8 1111010010
9 0110110000
10 0101111000
11 0100010000
```

✓ ساختار فایل **output.txt** به صورت زیر است و دو مثال از آن آورده شده است.

- **خط اول:** یک رشته شامل کلمه **YES** (وجود جواب) و **NO** (عدم وجود جواب) که با ویرگول از زمان اجرا جدا شده باشد نوشته شود

- **خطوط بعدی:** در صورت وجود جواب، مانند شکل سمت راست مقادیر هر نود نوشته شود. در صورتی که جوابی پیدا نشد چیزی نوشته نشود.

زمان اجرای برنامه

جواب داشتن یا نداشتن مسئله

حالت ۱ فایل output.txt در صورتی که مسئله جوابی ندارد

حالت ۲ فایل output.txt در صورتی که مسئله جواب دارد

```
output.txt
1 NO, 2ms

output.txt
1 YES, 2ms
2 0:Red
3 1:Green
4 2:Blue
5 3:Red
6 4:Yellow
7 5:Yellow
8 6:Cyan
9 7:Red
10 8:Blue
11 9:Red
```

رنگ های نود های گراف می باشد. در اینجا، کشور شماره ۰، رنگ قرمز و کشور شماره ۱، رنگ سبز شده است و ...

مهلت تحویل پروژه

۲۶ روز (تا ۳۱ خرداد ۱۴۰۰، ۱۲ شب)

نحوه ارسال پروژه

یک فایل زیپ به اسم و شماره دانشجویی تان شامل فایل های کد، **فایل گزارش**، **اسکرین شات برنامه**، دستورالعمل اجرا (چنانچه در محیطی غیر از ویژوال استودیو توسعه داده شود) باشد. این فایل در مدت زمان تحویل پروژه به آدرس amirmohammad.beuke@gmail.com ایمیل شود.

هر گونه سوالی داشتید می توانید با بنده از طریق ایمیل و دیگر راه ها در ارتباط باشید.

موفق باشید!

امیرمحمد بیوکی