

دانشگاه صنعتی امیر کبیر
(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

**تمرین اول برنامه‌نویسی
هوش مصنوعی و کاربردها**

پاییز ۹۸

توجه: پیش از شروع تمرین لطفا موارد زیر را با دقت مطالعه نمایید.

لطفا تمام فایل‌های تمرین را (از جمله فایل pdf گزارش و فایل‌های کد) در یک فایل zip/rar ذخیره کرده و نام آن را به PA1_XXXXXXXXX.zip تغییر دهید. سپس آن را در مدل بارگذاری کنید.

تمرین‌ها را می‌توانید با یکی از زبان‌های C++، Java یا Python انجام دهید.

پیاده سازی الگوریتم حل مسئله و کد توصیف مسئله باید مستقل از یکدیگر باشند؛ یعنی در صورت تغییر مساله نیازی به تغییر در الگوریتم‌ها نباشد.

برای هر سوال باید کدی جدا نوشته شود. برای مثال کدهای سوال ۳ بخش (a) را در فایل p3a.py ذخیره کنید. مهلت تحویل تمرین تا ۹ دی ۹۸ می‌باشد.

تمرین‌ها باید تک‌نفری انجام شوند و با هرگونه مشابهت در کدها برخورد خواهد شد.

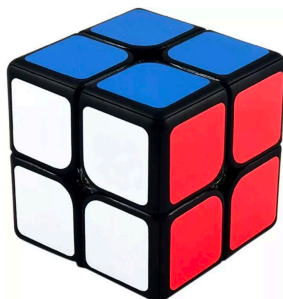
تمرین‌ها تحویل حضوری خواهند داشت. تاریخ تحویل حضوری متعاقبا اعلام خواهد شد.

در صورت داشتن هرگونه سوال به n3urotransmitter@gmail.com یا behzad.dara.99@gmail.com ایمیل دهید.

موفق باشید.

سوال اول: مکعب روییک

یک مکعب روبریک دو در دو را در ورودی گرفته و به وسیله الگوریتم‌های جست‌وجو آن را حل کنید. جابه‌جایی بین حالت‌ها (State) به همان صورتی است که در بازی واقعی انجام می‌شود و شما در هر مرحله می‌توانید یکی از سطوح را به صورت ساعت‌گرد یا پادساعت‌گرد بچرخانید و در کل ۱۲ انتخاب خواهید داشت. مکعب را به صورت زیر در نظر بگیرید:

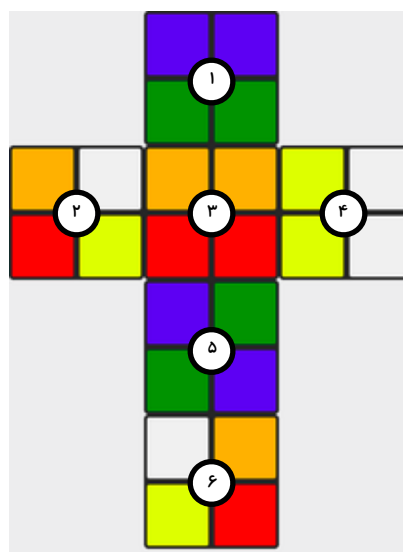


سطوح مکعب را به این ترتیب شماره گذاری میکنیم:

نارنجی = ۱ سبز = ۲ سفید = ۳ آبی = ۴ قرمز = ۵ زرد = ۶

برای چاپ حرکات انجام شده، شماره سطح مکعب و ساعت‌گرد یا پادساعت‌گرد بودن حرکت آنرا اعلام کنید. حالت اولیه مکعب به این صورت از ورودی گرفته می‌شود که به ترتیب شماره سطح‌ها که در بالا گفته شد، رنگ خانه‌های هر سطح را به ترتیب زیر وارد می‌کنیم. شماره‌ی رنگ‌ها هم به همان صورتی که در بالا گفته شده است می‌باشد. برای مثال برای شکل زیر، ورودی به صورت زیر خواهد بود:

4 4 2 2
1 3 5 6
1 1 5 5
6 3 6 3
4 2 2 4
3 1 6 5



پس از رسیدن به جواب، برنامه ی شما باید تعداد گره‌های تولید شده، تعداد گره‌های گسترش داده شده، عمق جواب و حداکثر تعداد گره‌هایی که در هنگام اجرا در حافظه قرار گرفتند را گزارش دهد.

مسئله باید با الگوریتم های زیر حل شود و جواب را بیابد:

۱. عمق اول با افزایش تدریجی عمق (عمق اولیه باید جزو پارامترهای قابل تغییر در برنامه شما باشد و عمق باید یکی یکی افزایش پیدا کند)

۲. دو جهت (دو طرف باید با سطح اول گرافی به سمت هم حرکت کنند)

۳. A* با تابع شهودی زیر

تعداد وجه با ۴ رنگ متفاوت ضرب در ۴

+

تعداد وجه با ۳ رنگ متفاوت ضرب در ۲

+

تعداد وجه با دو رنگ متفاوت

مثلا برای شکل صفحه‌ی قبل، تابع شهودی ۱۲ خواهد بود.

گزارش: فولدر تحویلی شما باید شامل یک فایل PDF باشد که در آن نحوه‌ی مدل کردن مسئله و پیاده‌سازی هر الگوریتم را آورده باشید. همچنین در گزارش خود باید تمامی الگوریتم‌ها را از نظر تعداد گره تولید شده، تعداد گره‌ی بسط داده شده، عمق جواب و حداکثر تعداد گره‌ی ذخیره شده در طی اجرا را با یکدیگر مقایسه کنید.

سوال دوم

در این سوال می‌خواهیم مسئله‌ی رنگ‌آمیزی گراف را حل کنیم.
فرض کنید می‌خواهیم نقشه‌ی استان‌های ایران را با ۴ رنگ، رنگ‌آمیزی کنیم به طوری که هیچ ۲ استان مجاورى رنگ مشابه نداشته باشند. خروجی نهایی مورد نظر ما مشابه تصویر زیر می‌باشد:



این مسئله یک مسئله‌ی NP-Complete می‌باشد. برای حل آن می‌توانیم از الگوریتم‌های بهینه‌سازی استفاده کنیم.

بخش الف) حل به کمک الگوریتم ژنتیک

در حالت کلی، مراحل حل این مسئله به کمک الگوریتم ژنتیک به صورت زیر است :

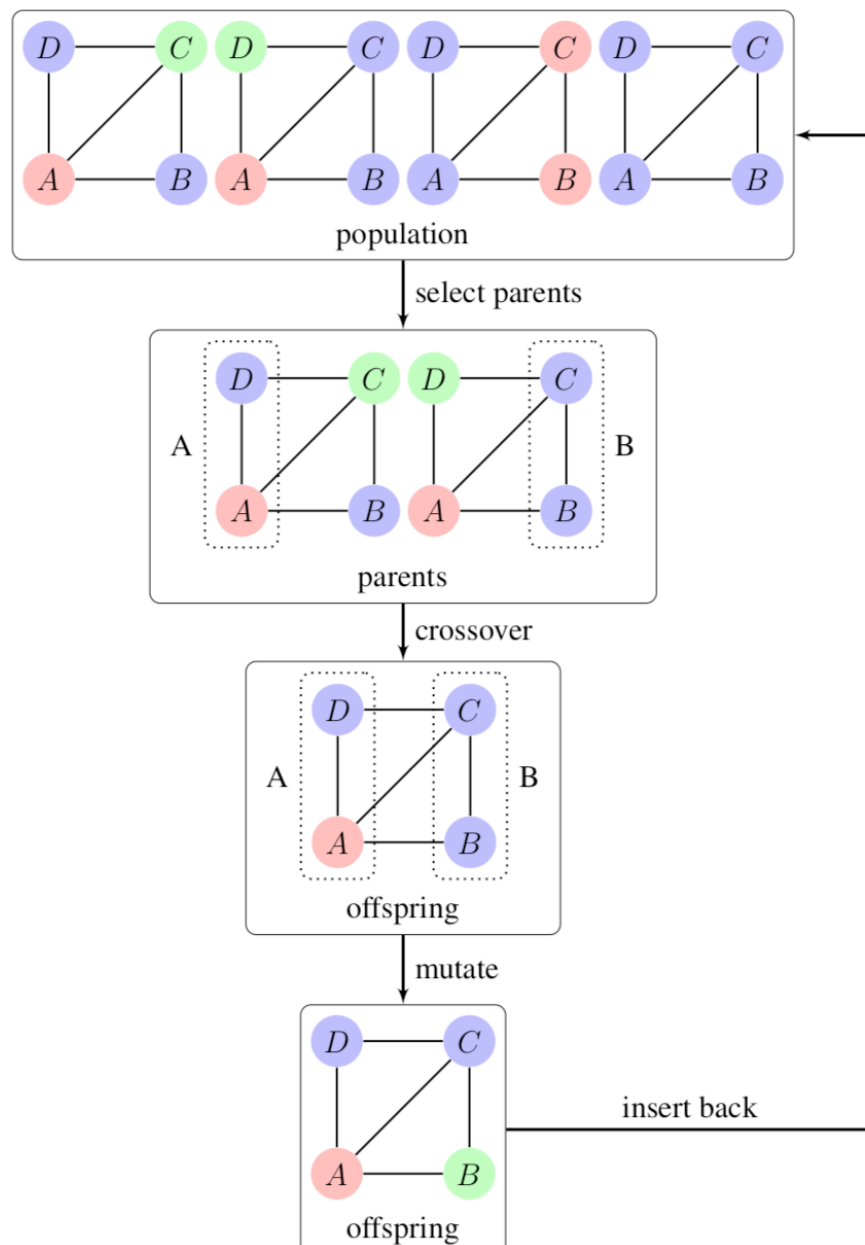


Figure 2.2: Example iteration of an evolutionary algorithm

جهت حل مسئله‌ی رنگ‌آمیزی گراف، مراحل زیر را دنبال کنید:

مرحله ۱ (تولید جماعت اولیه): جماعت اولیه را با population نام‌گذاری می‌کنیم. توجه کنید که تعداد جمعیت اولیه (populationSize) از پارامترهای قابل تنظیم مسئله می‌باشد. هریک از اعضای population، یک chromosome نام دارد. در ابتدا تمام گره‌های هریک از chromosome را به صورت رندوم، رنگ‌آمیزی کنید.

مرحله ۲ (تعیین میزان شایستگی هر chromosome): تابع fitness function، بیانگر میزان شایستگی هر chromosome در مسیر رسیدن به پاسخ مسئله می‌باشد که از آن برای تعیین والدین جهت تولید نسل بعدی استفاده می‌کنیم.

برای هریک از chromosome ها، تابع fitness function را به صورت زیر تعریف کنید:
فرض کنید $G = (V, E)$ گرافی با n گره و m یال باشد. اگر رنگ متعلق به گره‌ی i را با $c(i)$ نمایش دهیم، یک رنگ‌آمیزی گراف به صورت $C(G) = \{c(0), c(1), \dots, c(n)\}$ می‌باشد.
تابع δ را به صورت زیر تعریف می‌کنیم:

$$\text{for all } i, j \in E, \delta(i, j) = 1 \text{ if } c(i) \neq c(j) \text{ and } \delta(i, j) = 0 \text{ if } c(i) = c(j)$$

و با استفاده از این تابع، fitness function مورد نظر ما به صورت زیر تعریف می‌شود:

$$F(C(G)) = \frac{\sum_{i, j \in E} \delta(i, j)}{m}$$

برای هر chromosome، مقدار fitness را با توجه به این تابع به دست می‌آوریم.

مرحله ۳ (انتخاب والدین): در این مرحله از روش tournament selection استفاده می‌کنیم. در این روش تعداد k عضو را به صورت random انتخاب کرده و بهترین آن‌ها را برمی‌گزینیم. مثلاً اگر جماعت اولیه برابر ۱۰۰ باشد و k را برابر ۴ قرار دهیم، ۲۵ tournament اتفاق می‌افتد و ۲۵ عضو برگزیده می‌شوند. توجه کنید که k از پارامترهای قابل تنظیم می‌باشد و tournamentSize نام دارد. جماعت برگزیده، والدین نامیده می‌شوند.

مرحله ۴ (تولید نسل جدید): در مرحله‌ی بعد، باید تولید نسل جدید انجام گیرد. به این مرحله crossover گفته می‌شود. در این مرحله باید نسل جدید از طریق ترکیب هر ۲ والد برگزیده از مرحله‌ی قبل صورت گیرد. الگوریتم زیر، حالت ساده‌ای از تولید فرزندان را نشان می‌دهد:

```
for i = 1 to populationLength :  
    select x randomly from parents  
    select y randomly from parents where (y != x)  
    newChromosome[i] = crossover(x,y)  
end for;
```

در این مرحله، باید تابع crossover را تعریف کنیم.

روش‌های مختلفی برای تعریف تابع crossover وجود دارد. در اینجا ما هر عضو از population را به دو زیرگراف دلخواه تقسیم کرده به طوری که اشتراک این دو زیرگراف تهی و اجتماع این دو زیرگراف یک chromosome جدید را به وجود آورد.

مرحله ۵ (جهش): در این مرحله، تابع mutation را تعریف می‌کنیم. فرض کنید هر chromosome دارای تعداد n گرهی قابل رنگ‌آمیزی باشد. اصطلاحاً گفته می‌شود که هر کروموزوم، n ژن در اختیار دارد. همچنین پارامتر mutationRate را به عنوان یک پارامتر قابل تنظیم، تعریف می‌کنیم. تعداد ژن‌های جهش یافته از رابطه‌ی زیر به دست می‌آید:

$$\text{mutatedGenomes} = \text{populationLength} * n * \text{mutationRate}$$

و سپس به تعداد mutatedGenomes، ژن‌های موجود در population را به صورت رندوم تغییر می‌دهیم. در مثال رنگ‌آمیزی نقشه ایران، تعداد ۳۱ استان در اختیار داریم و فرض می‌کنیم جماعت اولیه را برابر ۱۰۰ در نظر گرفته‌ایم و mutationRate را برابر ۰.۰۲ قرار داده‌ایم. در این حالت رنگ ۶۲ ژن را به صورت رندوم تغییر می‌دهیم.

مرحله ۶ (بازگشت): در این مرحله، جماعت جدید ایجاد شده را به عنوان جماعت اولیه قرار می‌دهیم و کلیه مراحل را به اندازه‌ی متغیر numberOfGenerations که از پارامترهای قابل تنظیم مسئله می‌باشد، تکرار می‌کنیم.

از این الگوریتم جهت رنگ‌آمیزی نقشه‌ی ایران با تعداد ۴ رنگ، استفاده کنید.

مسئله را با تمام پارامترهای زیر، حل کنید و در نهایت درمورد تاثیر هر پارامتر در همگرایی پاسخ، بحث کنید.

```
numberOfGenerations = 50, 500, 5000
populationSize = 10, 100, 1000
tournamentSize = 2 for populationSize=10
tournamentSize = 2,5,10 for populationSize=100 and 1000
mutationRate = 0.01, 0.02, 0.05, 0.1
```

در هر generation، بهترین، بدترین و میانگین تابع شایستگی را ذخیره کرده و در نهایت نمودار میزان این ۳ متغیر در طول generations را نمایش دهید.

بخش ب) حل به کمک الگوریتم شبیه‌سازی ذوب فلزات

مسئله‌ی مذکور را با استفاده از الگوریتم شبیه‌سازی ذوب فلزات و با استفاده از تابع شایستگی ذکر شده در بخش الف، حل کنید. تابع کاهش دما را با روش‌های زیر پیاده‌سازی کرده و آن‌ها را مقایسه کنید.

(۱) $T_k = T_0 \alpha^k, \quad 0.8 \leq \alpha \leq 0.9$

(۲) $T_k = \frac{T_0}{1 + \alpha \text{Log}(1 + k)}, \quad \alpha > 1$

(۳) $T_k = \frac{T_0}{1 + \alpha k}, \quad \alpha > 0$

(۴) $T_k = \frac{T_0}{1 + \alpha k^2}, \quad \alpha > 0$