



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

تمرین برنامه‌نویسی

هوش مصنوعی

پاییز ۹۷

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 P_XXXXXXXXX.zip تغییر دهید. سپس آن را در مدل بارگذاری کنید.

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

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 ۳۰ آذر است.

برای هر تمرین علاوه بر کد پیاده سازی شده، گزارشی تهیه کنید که در آن نتایج به دست آمده در هر آزمایش ارائه و مقایسه شود.

در صورتی که گزارش درخواست شده برای هر سوال در فایل‌های ارسالی نباشد، نمره‌ای به آن سوال تعلق نمی‌گیرد.

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 nmoradzadehf@gmail.com یا ghaseemi.negin@gmail.com ایمیل دهید.

موفق باشید.

شرح پروژه

در این پروژه هر دانشجو می‌بایست مجموعه‌ای از الگوریتم‌های جستجوی کلاسیک که فهرست آنها در ادامه ذکر شده است را پیاده‌سازی و از این الگوریتم‌ها برای حل چند مساله جستجوی مختلف که در قالب یک واسط مشخص پیاده‌سازی می‌شوند استفاده کند.

پیاده‌سازی الگوریتم‌ها و مسائل مختلف می‌بایست به صورت مستقل از یکدیگر باشند به گونه‌ای که حل یک مساله جستجوی جدید نیازمند هیچ گونه تغییری در پیاده‌سازی الگوریتم‌های جستجو نباشند.

هر الگوریتم جستجو می‌بایست در انتهای اجرا، اطلاعات کافی در مورد اجرای الگوریتم مانند تعداد گره‌های مشاهده شده، تعداد گره‌های بسط داده شده، بهترین مسیر یافته شده، هزینه مسیر یافته شده، و حداکثر حافظه استفاده شده (بر حسب تعداد گره‌های نگهداری شده در حافظه) را بازگرداند.

توجه شود که گزارش کردن اطلاعات خواسته شده در هر مساله الزامی است و در صورت نبود گزارش نمره‌ای به آن سوال تعلق نمی‌گیرد.

الگوریتم‌ها

الگوریتم‌های که می‌بایست در این پروژه پیاده‌سازی کنید عبارتند از:

- سطح اول

- عمق اول (در سه حالت نامحدود، با عمق محدود و با افزایش تدریجی عمق)

- هزینه یکنواخت

- دو جهته

- *A

(الگوریتم‌های پیاده‌سازی شده بالا می‌بایست قابلیت اجرا در دو حالت جستجوی درختی یا گراف‌ی را دارا باشند).

- گرم و سرد کردن شبیه‌سازی شده (simulated annealing)

خروجی‌های الگوریتم:

- تعداد گره‌های ایجاد (مشاهده) شده

- تعداد گره‌های بسط داده شده

- راه حل و ارزش (شایستگی) راه حل یافته شده

• تپه نوردی (ساده، تصادفی، اولین انتخاب و شروع مجدد تصادفی)

خروجی‌های الگوریتم:

- تعداد گره‌های ایجاد (مشاهده) شده

- تعداد گره‌های بسط داده شده

- راه حل و ارزش (شایستگی) راه حل یافته شده

• الگوریتم ژنتیک

خروجی‌های الگوریتم:

- بهترین، بدترین و متوسط شایستگی در هر نسل

- تعداد نسل‌ها تا رسیدن به جواب بهینه (در صورت رسیدن به جواب بهینه)

الگوریتم ژنتیک را به گونه‌ای پیاده‌سازی کنید که اندازه جمعیت، تعداد کل ارزیابی‌های شایستگی (شرط خاتمه) و نرخ جهش آن قابل تنظیم باشد.

مسائل

به ازای هر یک از مسائلی که در ادامه معرفی می‌شود شما می‌بایست یک کلاس Problem بنویسید که در آن توابع لازم برای تعریف مساله (تابع حالت اولیه، عمل‌های ممکن در هر حالت، نتیجه هر عمل، آزمایش هدف، هزینه مسیر و گام، و تابع شهودی مورد استفاده) پیاده‌سازی شده باشند.

با اجرای الگوریتم‌های جستجوی ذکر شده برای هر مساله، عملکرد این الگوریتم‌ها در حل مساله مورد نظر را بررسی و با یکدیگر مقایسه کنید.

سوال اول

چهار زوج مختلف در یک سمت یک رودخانه قرار دارند. آن‌ها یک قایق در اختیار دارند و می‌خواهند به وسیله‌ی آن به آن سوی رودخانه بروند. در هر انتقال حداکثر ۲ و حداقل ۱ نفر می‌توانند در قایق قرار بگیرند و هیچکس نمی‌تواند بجز همسر خود با فردی از جنس مخالف در قایق تنها باشد!

این مسئله را با الگوریتم‌های زیر حل کرده و آن‌ها را از نظر تعداد گره‌های گسترش داده شده، تعداد گره‌های تولید شده، عمق جواب و حداکثر تعداد گره‌هایی که در طی الگوریتم در حافظه قرار می‌گیرند، مقایسه کنید.

(a) سطح اول

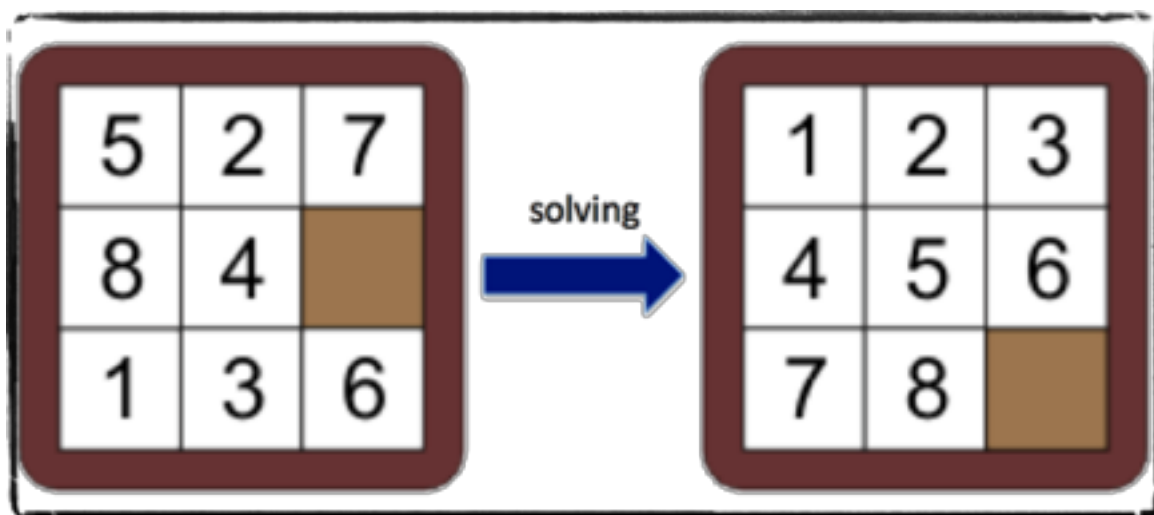
(b) دو جهته

(c) عمق اول (گرافی)

کد شما باید ترتیب انتقال افراد از این رودخانه را در خروجی نمایش دهد.

سوال دوم

مسئله‌ی sliding puzzle را در نظر بگیرید، در این مسئله یک مورد 3×3 پازل در اختیار داریم که یکی از خانه‌های آن خالی است. هدف این است که مسیر رسیدن به مورد با اعداد مرتب ۱ تا ۸ را از هر ورودی با قرارگیری رندوم پازل‌ها را پیدا کنیم.



این مسئله را با الگوریتم‌های زیر حل کرده و آن‌ها را از نظر تعداد گره‌های گسترش داده شده، تعداد گره‌های تولید شده، عمق جواب و حداکثر تعداد گره‌هایی که در طی الگوریتم در حافظه قرار می‌گیرند، مقایسه کنید.

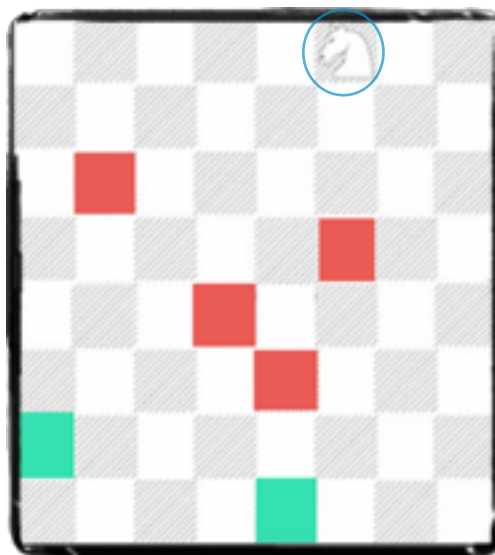
- (a) سطح اول
- (b) عمق اول (درختی)
- (c) عمق اول محدود شونده
- (d) A^* با تابع شهودی مجموع فاصله‌ی منتهن هر خانه با عددی که باید در آن خانه قرار بگیرد. (ارائه تابع شهودی بهتر نمره امتیازی دارد)

ورودی یک ماتریس 3×3 متشکل از اعداد صفر تا ۸ است که وضعیت شروع را مشخص می‌کند. کد شما باید حرکت‌های لازم برای رسیدن به حالت نهایی مطلوب را چاپ کند.

توجه کنید که مسئله ممکن است جواب نداشته باشد و کد شما باید این را تشخیص دهد. (امتیازی)

سوال سوم

یک صفحه‌ی 8×8 شطرنج در اختیار داریم. می‌خواهیم یک مهره‌ی اسب را از یک خانه به خانه‌های مشخصی انتقال دهیم. یک نمونه از این صفحه به صورت زیر است :



در این صفحه، اسب باید به یکی از خانه‌های سبز منتقل شود و همچنین اسب نمی‌تواند در خانه‌های قرمز قرار بگیرد.

این مسئله را با الگوریتم‌های زیر حل کرده و آن‌ها را از نظر تعداد گره‌های گسترش داده شده، تعداد گره‌های تولید شده، عمق جواب و حداکثر تعداد گره‌هایی که در طی الگوریتم در حافظه قرار می‌گیرند، مقایسه کنید.

(a) هزینه یکنواخت (هزینه هر حرکت را برابر ۱ در نظر بگیرید)

(b) عمق اول با افزایش تدریجی عمق

کد شما باید مسیر طی شده توسط اسب برای رسیدن به خانه‌های سبز را چاپ کند. ورودی یک ماتریس 8×8 خواهد بود که محل اسب با عدد ۱، خانه‌های خالی با عدد صفر، خانه‌های سبز با عدد ۲ و خانه‌های قرمز با عدد ۱- نمایش داده می‌شوند. توجه کنید که مسئله ممکن است جواب نداشته باشد و کد شما باید این را تشخیص دهد.

سوال چهارم: صف متعادل

یک صف به اندازه $3K$ در اختیار دارید. به وسیله الگوریتم تپه‌نوردی این صف را متعادل کنید. برای اندازه‌گیری تعادل صف، ابتدا اعضای صف را سه تا سه تا جدا می‌کنیم و سپس مجموع اعضا را در هر کدام از این سه قسمت اندازه‌گیری می‌کنیم. بیشینه مقدار این مجموع‌ها نشان‌دهنده مقدار عدم توازن است. هدف کمینه کردن این مقدار بیشینه است.

مثلا صف زیر را در نظر بگیرید:

2, 8, 6, 5, 2, 0

مقدار عدم توازن در این صف برابر می‌شود با

$$\text{Max}\{2+8+6, 5+2+0\} = \text{Max}\{16, 8\} = 16$$

با مرتب‌سازی این صف به این صورت به حالت بهینه می‌توانیم برسیم:

$$6, 5, 0, 2, 8, 2 \rightarrow \text{Max}\{6+5+0, 2+8+2\} = 12$$

برای این مساله یک تابع ارزش تعریف کنید و به وسیله الگوریتم‌های مختلف تپه‌نوردی (ساده، تصادفی، اولین انتخاب، شروع مجدد تصادفی) آن را به حالت بهینه برسانید.

در ورودی مساله عدد K را گرفته و به صورت رندم یک صف از اعداد صحیح به طول $3k$ تولید کنید و آن را به صورت بهینه مرتب کنید.

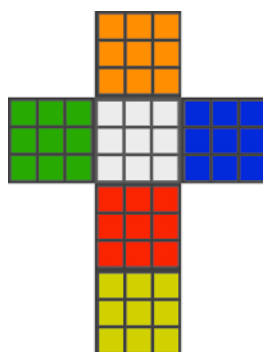
صف‌های رندم تولید شده و خروجی‌های هر کدام از الگوریتم‌ها را گزارش کرده و نتایج را مقایسه کنید.

سوال پنجم: مکعب روییک

یک مکعب سه در سه را در ورودی گرفته و به وسیله الگوریتم SA سعی کنید آنرا 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.

توجه شود که لزومی ندارد که الگوریتم بتواند مکعب را به طور کامل حل کند و بایستی به تعداد دفعاتی الگوریتم تکرار شود و در نهایت حالتی که به آن رسیده بازگردانده شود. علاوه بر حالت نهایی بایستی حرکاتی که در طی مسیر انجام می‌دهید را نیز اعلام کنید.

مکعب را به صورت زیر در نظر بگیرید:



سطوح مکعب را به این ترتیب شماره گذاری میکنیم:

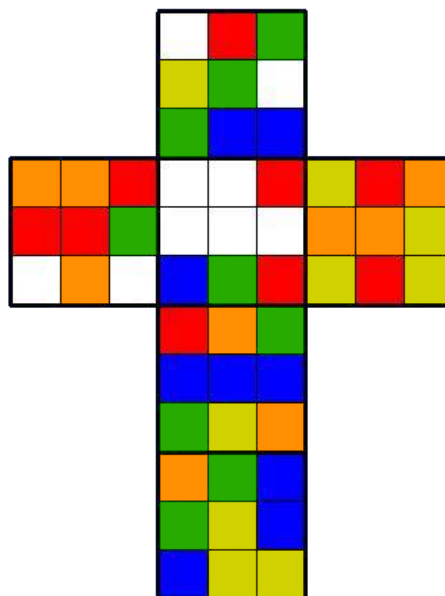
نارنجی = ۱ سبز = ۲ سفید = ۳ آبی = ۴ قرمز = ۵ زرد = ۶

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.

1	2	3
4	5	6
7	8	9

به طور مثال برای وارد کردن مکعب زیر به این صورت ورودی می‌دهیم:

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



این مساله را توسط الگوریتم SA حل کرده، سه روش مختلف برای کاهش دمای الگوریتم انتخاب کنید و با یکدیگر مقایسه کنید. خروجی‌ها و نتایج مقایسه را در گزارش بیاورید.

سوال ششم: رنگ آمیزی گراف

یک گراف و تعدادی رنگ داریم. می‌خواهیم گره‌های این گراف را طوری رنگ آمیزی کنیم که تا جای ممکن گره‌های همسایه رنگ یکسان نداشته باشند.

یک تابع ارزیابی تعریف کنید و این مساله را با الگوریتم ژنتیک حل کنید.

ورودی برنامه یک عدد که بیانگر حداکثر تعداد رنگ قابل استفاده است و یک گراف می‌باشد. گراف را به این صورت از ورودی بگیرید که ابتدا تعداد گره‌ها و سپس تعداد یال‌ها گرفته شود، سپس به تعداد یال‌ها دو عدد گرفته می‌شود که نشان‌دهنده گره‌های دو سر آن یال می‌باشد.

فایل [genetics.pdf](#) پیوست شده است. از آن برای چگونگی حل این مساله استفاده کنید.

موارد زیر را در گزارشتان بیاورید.

- نمودار بهترین، بدترین و متوسط شایستگی در هر نسل
- تاثیر کاهش یا افزایش احتمال جهش
- تاثیر کاهش یا افزایش تعداد برش‌ها
- تاثیر اندازه جمعیت بر همگرایی الگوریتم و کیفیت نتایج