

بسمه تعالی



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

تمرین اول برنامه نویسی درس هوش مصنوعی

هدف:

هدف این پروژه آشنایی شما با الگوریتم‌های جستجوی کلاسیک و نحوه فرموله کردن مسائل جستجوی مختلف در قالب یک رابط یکسان است.

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.

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
- دوجهته
- A^*

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

مسائل:

به ازاء هر یک از مسائلی که در این بخش 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.

مساله اول: ربات امدادگر

یک ربات امدادگر در یک محیط $n \times m$ خانه‌ای قرار دارد، ربات در ابتدا در نقطه ی $(1,1)$ قرار دارد و متوجه حضور یک مصدوم در نقطه ی (n,m) می‌شود، در این محیط دیوار برخی از خانه‌های مجاور غیرقابل عبور است و توسط آوار مسدود شده است و ربات می‌خواهد هر چه سریعتر خود را به مصدوم برساند.

مشخصات این آوار ها به صورت مختصات دو خانه مجاور مشخص می‌شود. به عنوان مثال در شکل زیر چهار دیوار به صورت زیر وجود دارد: (خط اول بیان می‌کند که از خانه ی $(3,2)$ نمی‌توان به خانه ی $(4,2)$ رفت چرا که مسیر توسط آوار مسدود شده)

3 2 4 2

3 3 4 3

2 3 2 4

3 3 3 4

S				
				G

ورودی : خط اول شامل دو عدد می باشد که به ترتیب n و m را مشخص می کنند.

در خط دوم تعداد دیوار ها مشخص می شود

و در ادامه به تعداد دیوار ها خط ورودی وجود دارد که هر خط شامل ۴ عدد پشت سر هم که با Space از همدیگر جدا شده اند

برای مثال برای شکل صفحه ی قبل ورودی بصورت زیر خواهد بود :

Input :

5 5

4

3 2 4 2

3 3 4 3

2 3 2 4

3 3 3 4

خروجی :

شامل یک دنباله از حرکات است که با نماد های $r u l d$ مشخص شده اند.

r	حرکت به راست
u	حرکت به بالا
l	حرکت به چپ
d	حرکت به پایین

برای حل مساله از الگوریتم های زیر استفاده نمایید:

- هزینه یکنواخت (هزینه حرکت از هر خانه به خانه مجاور آن را برابر یک در نظر بگیرید).
- عمق اول (گرافی)
- دوجهته
- A^* با تابع شهودی فاصله مستقیم

مساله دوم: پازل ۸ تایی

پازل ۸ تایی یک مساله کلاسیک در هوش مصنوعی است. این پازل یک مربع 3×3 است که یک خانه آن خالی و بقیه خانه‌ها اعداد ۱ تا ۸ را دارند. هدف مرتب کردن پازل با تغییر دادن جای خالی می‌باشد به صورتی که پازل در انتها به صورت زیر در بیاید:

	1	2
3	4	5
6	7	8

ورودی این برنامه یک ماتریس از نحوه قرارگیری اعداد جدول می‌باشد (به جای خانه خالی، عدد صفر قرار دهید)، به عنوان نمونه:

```
# sample input file(8-puzzle)

4 5 2
1 7 3
0 6 8
```

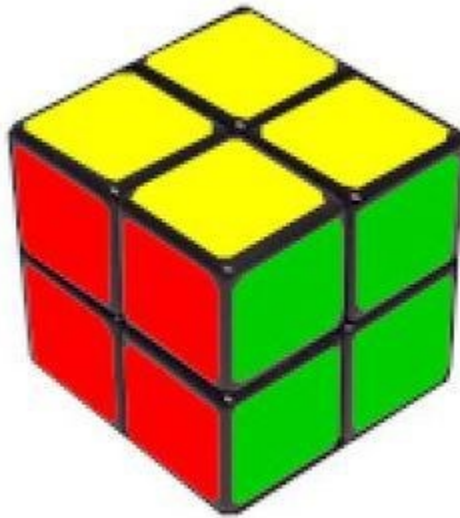
خروجی مساله همانند سوال یک دنباله ای از جهت ها می باشد.

برای حل مساله از الگوریتم های زیر استفاده نمایید:

- هزینه یکنواخت (هزینه حرکت از هر خانه به خانه مجاور آن را برابر یک در نظر بگیرید).
- عمق اول (گرافی)
- دوجهته
- A^* با تابع شهودی فاصله مستقیم

مساله سوم: مکعب روبیک ۲x۲

در این مساله شما باید یک راه حل برای حل مکعب روبیک که به عنوان ورودی مساله داده می شود پیدا کنید. برای سادگی مکعب مورد نظر ۲ در ۲ می باشد. خروجی شما باید شامل یک سری عمل متوالی باشد که با انجام آن از حالت ورودی به حالت مطلوب برسیم که همان حالتی است که قطعه های روی هر وجه مکعب همگی از یک رنگ باشند.



ورودی : شامل یک خط می باشد که هر رنگ با یک کاراکتر خاص مشخص شده و این رنگ ها با کاراکتر Space از هم جدا شده اند. در جدول زیر کاراکتر هر رنگ مشخص شده :

r	قرمز
g	سبز
b	آبی
w	سفید
o	نارنجی
y	زرد

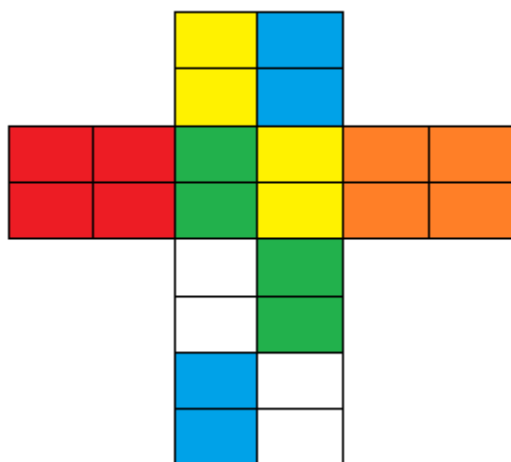
ترتیب مشخص کردن قطعه های روی هر وجه بصورت زیر می باشد :

		Top			
		1	2		
		3	4		
Front		5	6	Right	
17	18			21	22
19	20	7	8	23	24
		9	10		
		11	12		
		13	14		
		15	16		

به عنوان مثال ورودی زیر معادل مکعب مقابل می باشد :

Input :

y b y b g y g y w g w g b w b w r r r r o o o o



خروجی : شامل یک سری عملیات دوران است که در جدول زیر آمده است :

T	چرخاندن وجه بالایی (Top) در جهت عقربه های ساعت
TC	چرخاندن وجه بالایی (Top) در خلاف جهت عقربه های ساعت
F	چرخاندن وجه جلویی (Front) در جهت عقربه های ساعت
FC	چرخاندن وجه جلویی (Front) در خلاف جهت عقربه های ساعت
R	چرخاندن وجه راست (Right) در جهت عقربه های ساعت
RC	چرخاندن وجه راست (Right) در خلاف جهت عقربه های ساعت

توجه کنید که برای سادگی از یک سری حرکات صرف نظر شده است. برای مثال چرخاندن وجه چپ مکعب در جهت عقربه های ساعت دقیقا مشابه چرخاندن وجه راستی در خلاف جهت عقربه های ساعت می باشد ، لذا از آن صرف نظر شده است.

برای مثال یک نمونه از خروجی بصورت زیر می باشد :

Input : y b y b g y g y w g w g b w b w r r r r o o o o Output : R

با استفاده از الگوریتم های جستجوی زیر، مساله را حل کنید.

- سطح اول
- عمق اول با عمق محدود ۱۴ حرکت (می دانیم بیشترین تعداد حرکت برای حل روبیک ۲ در ۲، ۱۴ حرکت می باشد)
- عمق اول با افزایش تدریجی عمق

مواردی که در انجام این پروژه برنامه نویسی باید رعایت کنید:

- برنامه‌های خود را به یکی از زبان‌های جاوا، C++ یا پایتون پیاده سازی کنید.
- پروژه‌ها به صورت انفرادی تعریف شده است و در صورت تشابه جزئی یا کلی کدهای دو دانشجو، به هر دو نفر نمره صفر تعلق خواهد گرفت.
- برای هر تمرین علاوه بر کد پیاده سازی شده، گزارشی تهیه کنید که در آن نحوه مدل سازی مساله و خروجی‌های به دست آمده از هر روش جستجو ارائه شود و عملکرد الگوریتم‌ها با هم مقایسه شوند.
- در صورتی که گزارش درخواست شده در فایل های ارسالی نباشد، نمره ای تعلق نمی گیرد.
- کدها و مستندات خود را در یک فایل فشرده شده با قالب زیر نامگذاری و در سایت درس بارگذاری نمایید.

P1_Student#_Name.zip

P1_9431068_ArminKazemi.zip

- در صورت وجود هر گونه سوال و ابهام با ایمیل arminkz3@gmail.com در تماس باشید.