گزارش تمرین ۱ AI درس هوش مصنوعی

آریا بنائی زاده

۱) پیاده سازی الگوریتم‌ها: برای پیاده سازی از زبان C++ استفاده شده. الگوریتم BFS و دوجهته با استفاده از Queue و DFSها با استفاده از stack. الگوریتم ها به دو صورت گرافی و درختی پیاده سازی شده اند.

در مقایسه اولیه سرعت A* از بقیه الگوریتم‌ها بیشتر بود اما بیشترین مصرف حافظه را نیز دارا بود.

در مقابل BFS درختی بیشترین زمان را صرفنظر از الگوریتم‌های غیر کامل (DFS) و کمترین حافظه اشغالی را به خود اختصاص داد.

مسائل:

۱) پارچ آب:

در این مسئله ۶ عمل ممکن برای هر گره (A پر، B پر، A خالی و B خالی و A→B و B→A) داریم. مقدار آب در هر حالت از این اعمال بدست می آید. برای آن که در این مسئله مجبور به تولید گره های اضافی نشویم می توانیم از روش زیر استفاده کنیم (در کد نیز همین روش اعمال شده)

تعداد گره های گراف را برابر با حاصلضرب ظرفیت‌های A و B می گیریم (در این مسئله می شود ۶)

گره شماره n ام از رابطه B→capacity × A→water + B→water بدست می آید

حسن این روش این است که اگر عملی منجر به یک حالت تکراری شود ما آدرس و فضای حافظه‌ای خود از قبل داریم و تنها کافیست یال را بین این دو وصل کنیم.

روش درختی BFS حالت‌های بسیار زیادی تولید می کند (نتیجه حاصل از پیمایش در فایل ضمیمه آمده است)

برای جستجوی دوجهته کافیست از دو حالت <۰,۲> و <۰,۰> به طور همزمان شروع به پیمایش BFS کنیم

(صرفا باید توجه داشت اگر داریم از <۰,۲> یعنی حالت هدف پیمایش می‌کنیم اعمال را برعکس کنیم (مثلا پر کردن A در واقع خالی کردن آن را در گره مقابل نشان می‌دهد)

۲) مجموعه پازل ۸ تایی

BFS درختی: فضای حالت این مجموعه بسیار زیاد شد به طوری که بعد از مدت طولانی همچنان به جواب نرسید.

DFS عمق محدود ۸: با این روش با آرایه

$$\begin{Bmatrix} 5 & 1 & 2 \\ 3 & 4 & 0 \\ 6 & 7 & 8 \end{Bmatrix}$$

به جواب رسید اما این حالت بسیار ساده است و صرفا جای ۲ عدد جابجا شده اگر حالت پیچیده‌تر شود نمی‌توان به جواب رسید.

A^* : این روش از دو روش قبلی بهتر عمل کرده و با مقدار حالت تقریبا ... به جواب رسید

(آرایه ورودی :

8 4 6
5 0 7
2 3 1)

تابع شهودی بهتر : می توان هزینه رسیدن صفر به خانه مجاور عدد مورد نظر را نیز با تابع شهودی فاصله از هدف جمع کرد.

۳) ربات مسیریاب :

برای این مسئله نیز مانند مسئله قبل می توان از یک تابع برای نگاشت هر حالت به یک گره استفاده کرد.

در [illegible] حالت دوجهته از حالت شروع و هم حالت هدف شروع می کنیم و جایی که به هم می رسند نشان از یافته شدن مسیر است.

برای A^* و به دست آوردن تابع شهودی از یک تابع double استفاده شده تا فاصله مستقیم دقیق به دست بیاید

(تابع شهودی بهتر از تابع بالا جمع فاصله طول و عرض موقعیت ربات از هدف است)

از بین سه تابع هزینه یکنواخت، A^* دوجهته، A^* زمان کمتری داشت.

(تابع مورد استفاده در [illegible] برنامه یک آرایه ۲۵ × ۱۵ است)

* تمام خروجی کدها با فایل برنامه ضمیمه شده است.