به نام خدا

گزارش پروژه اول درس هوش مصنوعی

زهرا دهقانیان ۹۴۳۱۰۳۹

خرداد ۹۷

این پروژه در قالب دو پکیج جداگانه برای مسایل و الگوریتم طراحی شده . برای یکپارچه سازی قالب سوالات قابل حل توسط الگوریتم ها تمامی کلاس های Problem از یک کلاس problem (که در قبل از تعریف مسیله ها طراحی شد) ارث بری کرده اند .

برای هر سوال دو کلاس Node و Problem در نظر گرفته شده است. کلاس Problem مسیولیت دریافت ورودی و فراخوانی الگوریتم های مختلف برای حل مسیله را دارد . کلاس Node تمامی توابعی که در حل سوالات توسط الگوریتم ها ممکن است مورد نیاز شود را ، در خود جای می دهد .

برای مدل سازی باید ۵ مشخصه :

حالت اولیه = getFirstNode()

عمل های ممکن- هزینه مسیر - نتیجه هر عمل = getChild()

آزمایش هدف = isFinal()

در ادامه به بررسی نحوه مدل سازی و خروجی های به دست آمده از هر روش جستجو برای مسایل مختلف می پردازیم :

مساله اول :ربات امدادگر

۱) نحوه مدل سازی

این مسیله در قالب یک آرایه دوبعدی مدل شده است . توابع مورد نظر بدین صورت است

- **getFirstNode()**: خانه (۰,۰) به عنوان خانه شروع برمیگرداند.
- **getChild()** : این تابع به این صورت عمل می کند ، یک آرایه از نود ها که از این استیت قابل دستیابی است را برمیگرداند. در داخل این تابع بررسی میکنیم که ایا ۴ حرکت برای ما مجاز است یا نه (با بررسی ۴ شرط که به انتهای دیواره ها رسیدیم یا نه) و در صورت مجاز بودن این حالت به ارایه فرزندان اضافه شده و در نهایت این آرایه بازگردانده میشود.
- **isFinal()** : این تابع بررسی میکند ایا ۱ در خانه بایانی قرار دارد یا نه یعنی در **board** استیت فعلی در خانه ای با شماره **size** ورودی باشد .

۲) خروجی ها

خروجی ۴ الگوریتم به ازای ورودی روبرو به صورت است :

```
Output - AI_first_project (run)
run:
welcome to Robot problem ...
please enter your Maze :
5 5
4
3 2 4 2
3 3 4 3
2 3 2 4
3 3 4 3
```

```
This is A* search (graph):
this is count of expanded nodes : 37
this is count of absorved nodes : 48
and the memory usage : 49

***************
Answer is   D  D  D  D  R  R  R  R
BUILD SUCCESSFUL (total time: 5 seconds)
```

```
This is UniformCost search (graph) :
this is count of expanded nodes : 24
this is count of absorved nodes : 37
and the memory usage : 25

***************
Answer is   R  R  R  R  D  D  D  D
BUILD SUCCESSFUL (total time: 23 seconds)
```

```
This is DFS search (graph)
this is count of expanded nodes : 18
this is count of abserved nodes : 30
and the memory usage : 31
****************
Answer is :   R  R  R  R  D  D  R  D  D  R  R  R  R
BUILD SUCCESSFUL (total time: 2 seconds)
```

```
This is BDirectional search (graph) :
this is count of expanded nodes : 34
this is count of abserved nodes : 104
and the memory usage : 107
************
Answer is : R R R D D  D  D  D  R  R
BUILD SUCCESSFUL (total time: 3 seconds)
```

۳) مقایسه الگوریتم ها :

تمامی الگوریتم ها مسیر صحیحی پیدا کرده اند . الگوریتم دو جهته در بین الگوریتم ها بیشترین مصرف حافظه و بیشترین تعداد نود دیده شده را دارد.الگوریتم DFS با کمترین تعداد نود باز شده و دیده شده و حافظه مصرفی بهترین عملکرد را داشت .

مساله دوم :پازل ۸ تایی

۱) نحوه مدل سازی

این مسیله در قالب یک آرایه یک بعدی ۹ تایی مدل شده است. به عنوان تابع **heuristic** فاصله مستقیم (جمع فاصله طولی و عرضی از خانه مقصد) در نظر گرفته شده است

- **getFirstNode()**: این تابع ورودی دلخواه وارد شده توسط کاربر را به عنوان نود اولیه کامل میکند و میفرستد.
- **getChild()** : این تابع به این صورت عمل می کند ، یک آرایه از نود ها که از این استیت قابل دستیابی است را برمیگرداند. در داخل این تابع همانندمسیله قبل بررسی میکنیم که ایا ۴ حرکت برای ما مجاز است یا نه (با بررسی ۴ شرط که به انتهای دیواره ها رسیدیم یا نه) و در صورت مجاز بودن این حالت به ارایه فرزندان اضافه شده و در نهایت این آرایه بازگردانده میشود.
- **isFinal()** : این تابع از طریق یک متغیر **Boolean** بررسی میکند ایا در چینش بایانی قرار داریم با نه

۲) خروجی ها

خروجی ۴ الگوریتم به ازای ورودی روبرو به صورت است :

```
Output
Debugger Console    AI_first_project (run)
run:
welcome to eight puzzle prblem ...
please enter your puzzle :
3 1 2
6 4 5
7 8 0
```

```
This is A* search (graph):
this is count of expanded nodes : 4
this is count of abserved nodes : 7
and the memory usage : 8

***************
Answer is LLUU
BUILD SUCCESSFUL (total time: 2 seconds)
```

```
This is BDirectional search (graph) :
this is count of expanded nodes : 8
this is count of abserved nodes : 21
and the memory usage : 23
************
Answer is :  L L U U
BUILD SUCCESSFUL (total time: 2 seconds)
```

```
This is DFS search (graph)
this is count of expanded nodes : 32
this is count of absorved nodes : 58
and the memory usage : 59
****************
Answer is :  UULDDLUURDDLUURDDLUURDDLUURDDLUU
BUILD SUCCESSFUL (total time: 2 seconds)
```

```
This is UniformCost search (graph) :
this is count of expanded nodes : 10
this is count of absorved nodes : 20
and the memory usage : 16

***************
Answer is : LLUU
BUILD SUCCESSFUL (total time: 2 seconds)
```

۳) مقایسه الگوریتم ها :

از نظر مصرف حافظه و تعداد نود دیده شده الگوریتم dfs نسبت به بقیه ضعیف تر عمل کرده .

ضمن این که این الگوریتم مسیر بهینه را نیز بیدا نکرده (ولی جواب یافته شده صحیح است).

در این جا الگوریتم A* نسبت به بقیه یهتر عمل کرده است .

مساله سوم : مکعب روبیک ۲*۲

۱) نحوه مدل سازی

این مسیله در قالب یک آرایه یک بعدی ۲۴ تایی مدل شده است. در این مسیله با توجه به الگوریتم های انتخابی نیاز به **heuristic** نداریم . برای **ذخیره سازی اکشن هر مرحله از خانه اخر ارایه استفاده کردیم.**

- **getFirstNode()**: این تابع ورودی دلخواه وارد شده توسط کاربر را به عنوان نود اولیه کامل میکند و میفرستد.
- **getChild()** : این تابع به این صورت عمل می کند ، یک آرایه از نود ها که از این استیت قابل دستیابی است را برمیگرداند. در داخل این تابع ۶ حرکت موجود را انجام میدهد و به ارایه فرزندان اضافه میکند و در نهایت این آرایه بازگردانده میشود.
- **isFinal()** : این تابع با مقایسه هر ۴ خانه مجاور چک کیکند که در حالت پایانی قرار داریم

۲) خروجی ها

خروجی ۳ الگوریتم به ازای ورودی روبرو به صورت است :

```
Output ×
Debugger Console × AI_first_project (run) ×
run:
welcome to Rubik Problem :
please enter your Rubik :
w g w g
r w r w
b r b r
g b g b
o o o o
y y y y
```

```
This is IDDFS search (graph) :
***************
the answer was not found in depth 0
This is DLS search (graph) in depth :1
The Answer was found in depth : 1
this is count of expanded nodes : 1
this is count of abserved nodes : 5
and the memory usage : 6
***************
Answer is :  R
BUILD SUCCESSFUL (total time: 2 seconds)
```

```
This is BFS search (graph) :
this is count of expanded nodes : 5
this is count of abserved nodes : 21
and the memory usage : 22

***************
Answer is :  R


BUILD SUCCESSFUL (total time: 1 second)
```

```
This is DLS search (graph) in depth :14
The Answer was found in depth : 1
this is count of expanded nodes : 1
this is count of abserved nodes : 5
and the memory usage : 6
***************
Answer is :  R
BUILD SUCCESSFUL (total time: 2 seconds)
```

۳) مقایسه الگوریتم ها :

تمام الگوریتم ها جواب صحیح را یافتند .از نظر مصرف حافظه و تعداد نود دیده شده الگوریتم bfs نسبت به بقیه ضعیف تر عمل کرده . iddfs و dfs در این تست کیس تا حدودی مشابه عمل کرده اند (چون عمق جواب ۱ است) اما اگر عمق مسیله بیشتر شود متفاوت عمل میکنند .

پایان