**به نام خدا**

**گزارش پروژه اول درس هوش مصنوعی**

**زهرا دهقانیان 9431039**

**خرداد 97**

**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 5 مشخصه :**

**حالت اولیه = getFirstNode()**

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

**آزمایش هدف = isFinal()**

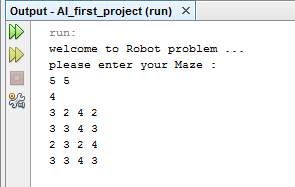
**در ادامه به بررسی نحوه مدل سازی و خروجی های به دست آمده از هر روش جستجو برای مسایل مختلف می پردازیم :**

**مساله اول: ربات امدادگر**

1. **نحوه مدل سازی**

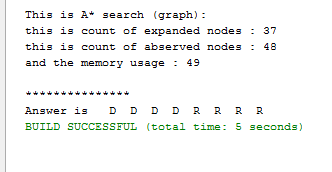
**این مسیله در قالب یک آرایه دوبعدی مدل شده است . توابع مورد نظر بدین صورت است**

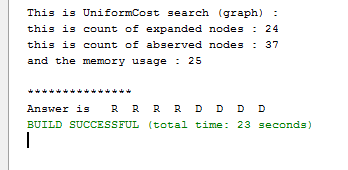
* **getFirstNode() : خانه (0.0) به عنوان خانه شروع برمیگرداند.**
* **getChild() : این تابع به این صورت عمل می کند ، یک آرایه از نود ها که از این استیت قابل دستیابی است را برمیگرداند. در داخل این تابع بررسی میکنیم که ایا ۴ حرکت برای ما مجاز است یا نه ( با بررسی ۴ شرط که به انتهای دیواره ها رسیدیم یا نه ) 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 .**

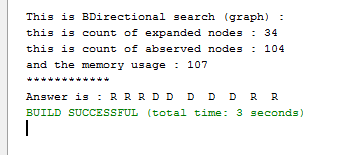
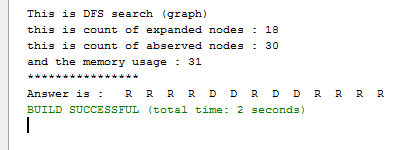
1. **خروجی ها**

**خروجی 4 الگوریتم به ازای ورودی روبرو به صورت**

**است :**

****





1. **مقایسه الگوریتم ها :**

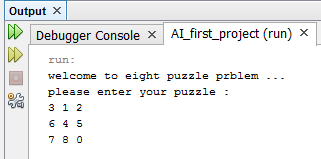
**تمامی الگوریتم ها مسیر صحیحی پیدا کرده اند . الگوریتم دو جهته در بین الگوریتم ها بیشترین مصرف حافظه و بیشترین تعداد نود دیده شده را دارد.الگوریتمDFS با کمترین تعداد نود باز شده و دیده شده و حافظه مصرفی بهترین عملکرد را داشت .**

**مساله دوم: پازل ۸ تایی**

1. **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**

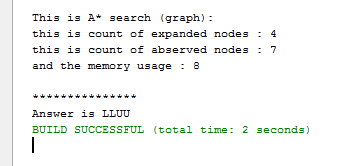
* **getFirstNode() : این تابع ورودی دلخواه وارد شده توسط کاربر را به عنوان نود اولیه کامل میکند و میفرستد.**
* **getChild() : این تابع به این صورت عمل می کند ، یک آرایه از نود ها که از این استیت قابل دستیابی است را برمیگرداند. در داخل این تابع همانندمسیله قبل بررسی میکنیم که ایا ۴ حرکت برای ما مجاز است 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**

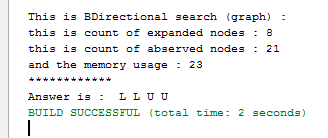


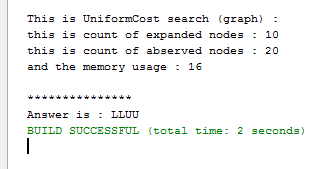
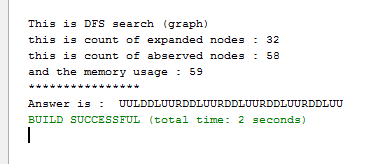
1. **خروجی ها**

**خروجی 4 الگوریتم به ازای ورودی روبرو به صورت**

**است :**







1. **مقایسه الگوریتم ها :**

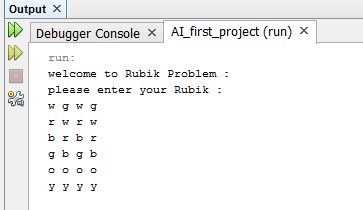
**از نظر مصرف حافظه و تعداد نود دیده شده الگوریتم dfs نسبت به بقیه ضعیف تر عمل کرده . ضمن این که این الگوریتم مسیر بهینه را نیز بیدا نکرده ( ولی جواب یافته شده صحیح است ). در این جا الگوریتم A\* نسبت به بقیه یهتر عمل کرده است .**

**مساله سوم: مکعب روبیک ۲\*۲**

1. **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.**

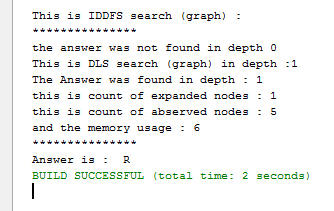
* **getFirstNode() : این تابع ورودی دلخواه وارد شده توسط کاربر را به عنوان نود اولیه کامل میکند و میفرستد.**
* **getChild() : این تابع به این صورت عمل می کند ، یک آرایه از نود ها که از این استیت قابل دستیابی است را برمیگرداند. در داخل این تابع ۶ حرکت موجود را انجام میدهد و به ارایه فرزندان اضافه میکند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

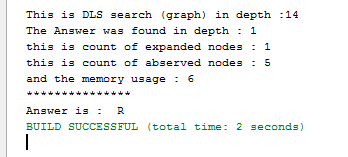
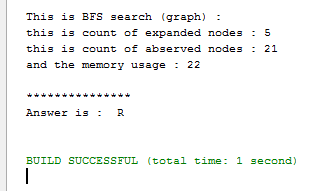


1. **خروجی ها**

**خروجی ۳ الگوریتم به ازای ورودی روبرو به صورت**

**است :**





1. **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 1 است) اما اگر عمق مسیله بیشتر شود متفاوت عمل میکنند .**

**پایان**