

به نام خدا

گزارش پروژه دوم 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 مسئله ها طراحی شد) ارث بری کرده اند .

برای هر سوال دو کلاس Node و Problem در نظر گرفته شده است. کلاس Problem مسئولیت دریافت ورودی و فراخوانی الگوریتم های مختلف برای حل مسئله 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 مشخصه های زیر مشخص باشد :

حالت اولیه = `getFirstNode()`

عمل های ممکن - نتیجه هر عمل = `getChild()`

هزینه مسیر = `getEval()`

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

مساله اول :رنگ آمیزی گراف

(۱) نحوه مدل سازی

این مسئله در قالب یک آرایه دوبعدی مدل شده است. این آرایه گراف ورودی را دریافت میکند. یک آرایه یک بعدی نیز برای رنگ هر راس در نظر گرفته شده است. توابع مورد نظر بدین صورت است

- **getNodeFirst()** تمام راس ها را با یک رنگ ، رنگ آمیزی میکند و به عنوان استیت اولیه برای تابع **hillClimbing** میفرستد
- **getNodeRandom()** : این تابع هر راس را با یک رنگ تصادفی رنگ میکند .
- **getChild()** : این تابع به این صورت عمل می کند ، عدد مربوط به رنگ هر راس را یک واحد زیاد میکند و در صورت غیر مجاز شدن رنگ راس ، به محدوده مجاز تبدیل میکند(با مود گرفتن به تعداد رنگ ها)
- **getEval()** : این تابع به ازای هر دو راسی که به یکدیگر یال دارند و هم رنگ هستند مقدار **eval** را یک واحد زیاد کرده و در نهایت به عنوان ارزش این استیت برمیگرداند.

```
Output - AI_2ndProject (run) x
run:
Welcome to Graph Coloring Problem
Enter number of verticesz
3
Enter matrix
0 1 1
0 0 1
0 0 0
Enter number of colors
4
```

(۲) خروجی ها

خروجی الگوریتم به ازای ورودی روبرو به صورت است :

```
result of simple HillClimbing with Eval : 1
count of seenNodes : 6
count of steps : 2
Answer is : 2 1 1
*****
```

```
result of stochastic HillClimbing with Eval : 1
count of seenNodes : 6
count of steps : 2
Answer is : 1 2 1
*****
```

```
result of firstChoice HillClimbing with EVal : 1
count of seenNodes : 6
count of steps : 2
Answer is : 2 1 1
*****
```

```
result of Random restart HillClimbing with EVal : 0
count of seenNodes : 15
count of steps : 5
Answer is : 2 0 1
*****
```

۳) مقایسه الگوریتم ها :

تنها الگوریتم با شروع تصادفی جواب صحیح پیدا کرده است . این الگوریتم بیشترین نود دیده شده و بیشترین تعداد مراحل را نیز به خود اختصاص داده که به معنی مصرف حافظه بیشتر است

مساله دوم :جدول حروف

(۱) نحوه مدل سازی

این مسئله در قالب یک آرایه دو بعدی مدل شده است. برای نگهداری دیکشنری کلمات نیز از یک آرایه یک بعدی استفاده شده است .

- **getNode()**: این تابع جدول اولیه کاربر را به عنوان نود اولیه ایجاد و ارسال میکند.
- **getChild()**: این تابع به این صورت عمل می کند ، که جایگشت های دو خانه مجاور را به ترتیب به عنوان یک جدول جدید تولید میکند و در آرایه ای ذخیره کرده و در نهایت ارسال میکند .
- **getEval()**: این تابع ارزیابی به ازای هر کلمه ای که در دیکشنری وجود دارد ، در صورتی که دوخانه پشت سر هم دو حرف پشت هم از کلمه دیکشنری باشد ، **eval** را یک واحد زیاد میکند. (**eval** بیشتر بهتر است)

در نهایت در تابع **simulatedAnnealing** تا رسیدن به دمای + میچرخیم و جواب یافته شده را چاپ میکنیم .

```
Output - AI_2ndProject (run) x
run:
welcome to word table problem
enter dimansion
3
enter table
e r t
e h l
e i d
enter number of word in dictionary
3
enter dictionary
ret
hel
tli

result of simmulated annealing with fix Teperature ONE
match percentage : 100%
count of seenNodes : 396
count of steps : 44
ret
hel
edi
*****
```

(۲) خروجی ها

خروجی الگوریتم به ازای ورودی روبرو به صورت است :

```
result of simmulated annealing with fix Teperature TWO
match percentage : 0%
count of seenNodes : 9
count of steps : 1
ert
ehl
eid
*****
```

```
result of simulated annealing with fix Teperature THREE
match percentage : 80%
count of seenNodes : 9
count of steps : 1
ert
hel
eid
*****
```

(۳) مقایسه الگوریتم ها :

از نظر مصرف حافظه و تعداد نود دیده شده اولین نوع کم کردن حافظه مصرف بیشتری داشته اما به همین نسبت در پیدا کردن جواب موفق عمل کرده و جواب دقیق تری بدست آورده است . به نظر می رسد در صورت داشتن حافظه بیشتر ، رویه اول بهتر بوده و در صورت کمبود حافظه از رویه های دیگر می توان استفاده کرد .

پایان