هوش مصنوعی

گزارش پروژه اول

حمیدرضا خدادادی 810197499

هدف:

ما در این پروژه مسئله ای را طرح کرده و قصد داریم که با استفاده از الگوریتم های آگاهانه و ناآگاهانه راه حل هایی برای حل مسئله ی طرح شده ، پیدا کنیم. سپس این الگوریتم های پیدا کرده را با هم مقایسه کرده و ویژگی های هر یک را به نسبت به دیگری می سنجیم.

کلید واژه:

مار – دانه – BFS - IDS - A* - Weighted A* - heuristic - consistent - admissible

خلاصه:

ما یک صفحه دو بعدی و یک مار با اندازه ی یک خانه از این صفحه داریم ، تعدادی دانه ی یک امتیازی و تعدادی دانه ی دو امتیازی در این صفحه قرار دارد ، مار با خوردن هر دانه و پس از خروج از آن خانه ، یک واحد رشد می کند ، و دانه ها پس از خورده شدن از صفحه ی بازی حذف می شوند و اینکه دانه های دو امتیازی پس از دو بار خورده شدن از صفحه ی بازی حذف می شوند ، مار در چهار جهت جغرافیایی می تواند حرکت کند به شرط اینکه سر مار بدنش را قطع نکند ، و مار می تواند از صفحه ی مختصات خارج شده و از طرف دیگر وارد صفحه ی بازی شود. هدف ما پیدا کردن بهینه ترین راه از لحاظ تعداد گام ها و زمان است.

ما یک node تعریف کردیم که state و نوع اکشن و هزینه ی مسیر و پدر آن نود را نگه می دارد. و state را این گونه تعریف کردیم که مختصات دانه های باقی مانده در صفحه و مختصات فعلی بدن مار را نگه می دارد.

اکشن ها ، چهار جهت جغرافیایی ، یعنی بالا ، پایین ، چپ و راست تعریف شده است.

استیت اولیه بدین گونه است که همه ی دانه ها در صفحه ی بازی هستند و مار با طول یک ، در مختصات اولیه ی خود قرار دارد.

استیت هدف بدین گونه است که همه ی دانه ها توسط مار خورده شده باشد و از صفحه ی بازی حذف شده اند و مار با طول و مختصات نهایی خود ، در صفحه ی بازی قرار دارد.

تابع check_border ، مختصات را می گیرد و اگر از صفحه ی بازی خارج شده باشد ، مختصات جدید و در طرف دیگر صفحه را به خروجی می دهد.

تابع check_seed_existence ، حالت های وجود داشتن دانه یک امتیازی و دو امتیازی و وجود نداشتن دانه در خانه ی هدف را بررسی می کند و مختصات بدن مار و وضعیت دانه ها را پس از رفتن سر مار به خانه ی جدید ، به خروجی می دهد.

تابع check_collision ، حالت های مختلف برخورد سر مار با بدنش را بررسی می کند و در صورت برخورد از رفتن مار به خانه

هوش مصنوعی

گزارش پروژه اول

حمیدرضا خدادادی 810197499

ی جدید جلوگیری می کند.

تابع new_state ، استیت فعلی و اکشن بعدی را می گیرد و با استفاده از تابع های بالا ، استیت جدید را تولید می کند.

الگوریتم BFS:

ما در این الگوریتم ، برای frontier از ساختار لیست و برای explored از ساختار ست استفاده کرده ایم.

فرانتیر به صورت صف مورد استفاده قرار می گیرد.

به ازای هر استیت ، وضعیت حرکت در هر چهار جهت را بررسی کرده و استیت های جدید را می سازیم.

این استیت ها اگر در اکسپلورد نباشند را به آن اضافه کرده و نود مربوط به این استیت را به فرانتیر اضافه می کنیم.

در این الگوریتم هزینه ی اجرای هر گام یک است پس در نتیجه جواب بدست آمده بهینه خواهد بود.

هزینه ی زمانی اجرای الگوریتم: O(b^d)
هزینه ی حافظه اجرای الگوریتم: O(b^d)

این الگوریتم از لحاظ هزینه ی حافظه ، پر هزینه است.

```
BFS Algoritm:
**********
Test: 1
Path cost:  12
Path:  ['D', 'L', 'U', 'U', 'L', 'U', 'L', 'L', 'U', 'U', 'L', 'L']
Time:  0.09375 s
States Number:  14431
Separate states Number:  6578
##########
**********
Test: 2
Path cost:  15
Path:  ['U', 'R', 'D', 'L', 'L', 'U', 'U', 'U', 'U', 'L', 'U', 'L', 'L', 'L', 'L']
Time:  1.140625 s
States Number:  123397
Separate states Number:  54215
##########
**********
Test: 3
Path cost:  25
Path:  ['U', 'R', 'D', 'D', 'D', 'R', 'D', 'R', 'R', 'D', 'D', 'R', 'R', 'R', 'U', 'R', 'R', 'D', 'L', 'L', 'L', 'U', 'U', 'L', 'L']
Time:  7.71875 s
States Number:  635961
Separate states Number:  275113
##########
```

هوش مصنوعی

گزارش پروژه اول

حمیدرضا خدادادی 810197499

الگوریتم IDS:

ما در این الگوریتم ، برای frontier از ساختار لیست و برای explored از ساختار دیکشنری استفاده کرده ایم.

فرانتیر به صورت استک مورد استفاده قرار می گیرد.

به ازای هر استیت ، وضعیت حرکت در هر چهار جهت را بررسی کرده و استیت های جدید را می سازیم.

این استیت ها اگر در اکسپلورد نباشند و یا باشند ولی هزینه ی استیت فعلی از هزینه ی استیت قبلی کمتر باشد ، آن استیت قبلی را دیگر در نظر نگرفته و این استیت جدید را به آن اضافه کرده و نود مربوط به این استیت را به فرانتیر اضافه می کنیم.

هر بار تابع DFS را با یک بیشترین عمق صدا می کنیم و این بیشترین عمق تا زمانی که جواب به دست بیاید ، افزایش می یابد.

در این الگوریتم هزینه ی اجرای هر گام یک است پس در نتیجه جواب بدست آمده بهینه خواهد بود.

هزینه ی زمانی اجرای الگوریتم: b^1 + b^2 + … + b^d
هزینه ی حافظه اجرای الگوریتم: O(b*d)

این الگوریتم از لحاظ هزینه ی زمانی ، پر هزینه است.

```
IDS Algoritm:
**********
Test: 1
Path cost:  12
Path:  ['L', 'D', 'L', 'D', 'R', 'U', 'L', 'U', 'U', 'L', 'U', 'L']
Time:  0.75 s
States Number:  74590
Separate states Number:  12126
##########
**********
Test: 2
Path cost:  15
Path:  ['R', 'U', 'L', 'L', 'D', 'L', 'U', 'U', 'U', 'U', 'L', 'L', 'L', 'L', 'U']
Time:  8.890625 s
States Number:  814010
Separate states Number:  101891
##########
**********
Test: 3
Path cost:  25
Path:  ['R', 'U', 'R', 'D', 'D', 'D', 'R', 'R', 'D', 'R', 'R', 'R', 'D', 'D', 'R', 'R', 'U', 'L', 'L', 'D', 'L', 'L', 'L', 'U', 'U']
Time:  63.828125 s
States Number:  5642372
Separate states Number:  683185
##########
```

هوش مصنوعی

گزارش پروژه اول

حمیدرضا خدادادی 810197499

الگوریتم *A و *A wighted:

ما در این الگوریتم ، برای frontier از ساختار لیست و برای explored از ساختار دیکشنری استفاده کرده ایم.

فرانتیر به صورت استک مورد استفاده قرار می گیرد. برای فرانتیر از ساختار heapq استفاده کرده ایم.

به ازای هر استیت ، وضعیت حرکت در هر چهار جهت را بررسی کرده و استیت های جدید را می سازیم.

این استیت ها اگر در اکسپلورد نباشند و یا باشند ولی هزینه تخمینی استیت فعلی از هزینه ی تخمینی استیت قبلی کمتر باشد ، آن استیت قبلی را دیگر در نظر نگرفته و این استیت جدید را به آن اضافه کرده و نود مربوط به این استیت را به فرانتیر اضافه می کنیم.

هیوریستیک اول تعداد دانه های باقی مانده در صفحه خواهد بود که هم ادمیسبل است و هم کانسیستنت است. پس بهینه است.

در حالت واقعی ، در هر حرکت فقط و ماکسیمم یک دانه را می شود خورد ، پس هزینه ی واقعی بزرگتر مساوی با تعداد دانه ها خواهد بود. پس ادمیسبل است. و می دانیم هزینه ی هر اکشن برابر با یک است. (هزینه واقعی جا به جایی بین دو نود) و تفاوت هیوریستیک به ازای هر دوتا استیت یا صفر است و یا یک. یعنی اگر دانه را بخورد یک است وگرنه صفر است. پس تفاوت هیوریستیک بین دو نود ، کمتر مساوی با هزینه ی واقعی آن جا به جایی خواهد بود.

هیوریستیک دوم فاصله ی سر مار از دورترین دانه است که ادمیسبل است.

در حالت واقعی ، باید مار به خانه ی هر دانه برسد تا آن را بخورد ، یعنی سر مار حداقل باید به دور ترین دانه برسد ، پس هزینه ی واقعی آن بزرگتر مساوی با فاصله ی سر مار از دور ترین دانه است. پس ادمیسبل است.

در ادامه ما ، هیوریستیک تعداد دانه های باقی مانده که بهینه است را در دو وزن 2.3 و 5 ضرب می کنیم.

این کار باعث می شود هزینه زمانی الگوریتم *A کاهش یابد ولی ممکن است ما را از جواب بهینه دور کند.

هر چه وزن بیشتری را در هیوریستیک ضرب کنیم ، زمان الگوریتم کمتر می شود. در عکس آخر مشاهده می کنید که هزینه ی زمانی کم شده ولی جواب بدست آمده بهینه نیست.

هوش مصنوعی

گزارش پروژه اول

حمیدرضا خدادادی 810197499

هیوریستیک اول و دوم در الگوریتم A*:

```
A star Algoritm:
Heuristic: 1
**********
Test: 1
Path cost:  12
Path:  ['D', 'L', 'U', 'U', 'L', 'L', 'U', 'L', 'U', 'L', 'U', 'L']
Time:  0.09375 s
States Number:  8152
Separate states Number:  4254
##########
**********
Test: 2
Path cost:  15
Path:  ['U', 'R', 'D', 'L', 'L', 'U', 'U', 'U', 'L', 'U', 'U', 'L', 'L', 'L', 'L']
Time:  0.84375 s
States Number:  65800
Separate states Number:  32032
##########
**********
Test: 3
Path cost:  25
Path:  ['U', 'R', 'D', 'D', 'D', 'R', 'D', 'R', 'R', 'D', 'R', 'R', 'D', 'R', 'R', 'R', 'U', 'L', 'L', 'D', 'L', 'L', 'U', 'U', 'L']
Time:  4.890625 s
States Number:  351309
Separate states Number:  160276
##########
```

```
Heuristic: 2
**********
Test: 1
Path cost:  12
Path:  ['L', 'D', 'L', 'D', 'R', 'U', 'L', 'U', 'U', 'L', 'U', 'L']
Time:  0.0625 s
States Number:  4811
Separate states Number:  2764
##########
**********
Test: 2
Path cost:  15
Path:  ['U', 'R', 'D', 'L', 'L', 'U', 'U', 'L', 'U', 'U', 'L', 'L', 'L', 'U', 'L']
Time:  0.125 s
States Number:  8755
Separate states Number:  5306
##########
**********
Test: 3
Path cost:  25
Path:  ['U', 'R', 'D', 'D', 'D', 'R', 'R', 'R', 'D', 'R', 'D', 'R', 'D', 'R', 'U', 'R', 'R', 'D', 'L', 'L', 'L', 'L', 'U', 'U', 'L']
Time:  1.453125 s
States Number:  97161
Separate states Number:  47133
##########
```

هوش مصنوعی

گزارش پروژه اول

حمیدرضا خدادادی 810197499

هیوریستیک اول (بهینه) با وزن های 2.3 و 5:

```
Wighted A star Algoritm:
**********
Test: 1
Path cost:  12
Path:  ['D', 'L', 'U', 'L', 'U', 'U', 'L', 'L', 'L', 'L', 'U', 'U']
Time:  0.03125 s
States Number:  3220
Separate states Number:  1837
##########
**********
Test: 2
Path cost:  15
Path:  ['R', 'U', 'L', 'D', 'L', 'U', 'L', 'U', 'U', 'U', 'U', 'L', 'L', 'L', 'L']
Time:  0.1875 s
States Number:  14322
Separate states Number:  8587
##########
**********
Test: 3
Path cost:  25
Path:  ['U', 'R', 'D', 'D', 'R', 'D', 'R', 'R', 'D', 'D', 'D', 'R', 'R', 'R', 'R', 'R', 'U', 'L', 'L', 'D', 'L', 'U', 'L', 'L', 'U']
Time:  1.703125 s
States Number:  135063
Separate states Number:  65064
##########
```

```
**********
Test: 1
Path cost:  15
Path:  ['U', 'L', 'L', 'U', 'L', 'L', 'U', 'L', 'L', 'U', 'L', 'L', 'L', 'L', 'L']
Time:  0.046875 s
States Number:  3757
Separate states Number:  1577
##########
**********
Test: 2
Path cost:  17
Path:  ['U', 'L', 'D', 'R', 'R', 'U', 'L', 'L', 'U', 'L', 'U', 'U', 'L', 'L', 'L', 'U', 'L']
Time:  0.046875 s
States Number:  3560
Separate states Number:  2403
##########
**********
Test: 3
Path cost:  26
Path:  ['U', 'R', 'D', 'D', 'D', 'R', 'D', 'R', 'R', 'R', 'U', 'L', 'D', 'D', 'D', 'R', 'R', 'R', 'R', 'U', 'R', 'D', 'D', 'L', 'L', 'U']
Time:  0.53125 s
States Number:  45030
Separate states Number:  23316
##########
```

هوش مصنوعی

گزارش پروژه اول

حمیدرضا خدادادی 810197499

Test 1	فاصله ی جواب	مسیر جواب	تعداد استیت دیده شده	تعداد استیت دیده مجزای شده	میانگین زمان اجرا
BFS	12	D-L-U-U-L-U-L-L-U-U-L-L	14431	6578	0.1015625 s
IDS	12	L-D-L-D-R-U-L-U-U-L-U-L	74590	12126	0.7578125 s
A* Heuristic 1	12	D-L-U-U-L-L-U-L-U-L-U-L	8152	4254	0.09375 s
A* Heuristic 2	12	L-D-L-D-R-U-L-U-U-L-U-L	4811	2764	0.0625 s
Wighted A* Heuristic 1 $\alpha = 2.2$	12	D-L-U-L-U-U-L-L-L-U-U	3220	1837	0.03125 s
Wighted A* Heuristic 1 $\alpha = 5$	15	U-L-L-U-L-L-U-L-L-U-L-L-L-L-L	3757	1577	0.0390625 s

Test 2	فاصله ی جواب	مسیر جواب	تعداد استیت دیده شده	تعداد استیت دیده مجزای شده	میانگین زمان اجرا
BFS	15	U-R-D-L-L-U-U-U-U-L-U-L-L-L-L	123397	54215	1.1640625 s
IDS	15	R-U-L-L-D-L-U-U-U-U-L-L-L-L-U	814010	101891	9.0234375 s
A* Heuristic 1	15	U-R-D-L-L-U-U-U-L-U-U-L-L-L-L	65800	32032	0.84375 s
A* Heuristic 2	15	U-R-D-L-L-U-U-L-U-U-L-L-L-U-L	8755	5306	0.1328125 s
Wighted A* Heuristic 1 $\alpha = 2.2$	15	R-U-L-D-L-U-L-U-U-U-U-L-L-L-L	14322	8587	0.1953125 s
Wighted A* Heuristic 1 $\alpha = 5$	17	U-R-D-D-D-R-D-R-R-R-U-L-D-D-D-R-R-R-R-U-R-D-D-L-L-U	3560	2403	0.046875 s

هوش مصنوعی

گزارش پروژه اول

حمیدرضا خدادادی 810197499

Test 1	فاصله ی جواب	مسیر جواب	تعداد استیت دیده شده	تعداد استیت دیده مجزای شده	میانگین زمان اجرا
BFS	25	U-R-D-D-D-R-D-R-R-D-D-R-R-R-U-R-R-D-L-L-L-U-U-L-L	635961	275113	7.78125 s
IDS	25	R-U-R-D-D-D-R-R-D-R-R-R-D-D-R-R-U-L-L-D-L-L-L-U-U	5642372	683185	65.94531 s
A* Heuristic 1	25	U-R-D-D-D-R-D-R-R-D-R-R-D-R-R-R-U-L-L-D-L-L-U-U-L	351309	160276	4.921875 s
A* Heuristic 2	25	U-R-D-D-D-R-R-R-D-R-D-R-D-R-U-R-R-D-L-L-L-L-U-U-L	97161	47133	1.4375 s
Wighted A* Heuristic 1 $\alpha = 2.2$	25	U-R-D-D-R-D-R-R-D-D-D-R-R-R-R-R-U-L-L-D-L-U-L-L-U	135063	65064	1.7109375 s
Wighted A* Heuristic 1 $\alpha = 5$	26	U-R-D-D-D-R-D-R-R-R-U-L-D-D-D-R-R-R-R-U-R-D-D-L-L-U	45030	23316	0.546875 s