#### «به نام خدا»

### گزارش پروژه ۱ درس مبانی هوش مصنوعی

## مهدی شاهینی ۹۹۲۳۰۴۰

نکته: در الگوریتمهای ۲ و ۳ زمانی که چندبار ران شد چند جواب مختلف در مدت زمان حدودی یکسان بدست آمد که همگی جواب صحیح بودند لطفا هنگام چک کردن صحت پاسخها گرافیک پاسخ را نیز مشاهده کنید.

#### گزارش ۱:

فعالیت ۴، در واقع چرخش ۹۰ درجه ساعت گرد صفحه بالایی روبیک است. بنابراین باید ماتریسی که به این ناحیه اشاره می کند ۹۰ درجه بچرخد. دو ردیف اول ماتریس state به این صفحه اشاره می کند که ۲ ردیف اول با دستور rot90 می چرخد حال باید خانههای جانبی که تغییر کردهاند را بروزرسانی کنیم. برای این کار چون هر دو ردیف یک مربع را نشان می دهند در واقع با چرخ ۹۰ درجه هر ردیف دو ردیف شیفت می خورد بنابراین ردیف ۲ به ردیف استارت ردیف استارت به ردیف ۸ و سایر ردیفها دو ردیف شیفت خورده یعنی ۶ به ۴ و ۴ به ۲ تبدیل می شود. علت این شیفت خوردن این است که در واقع از هر مربع به مربع مجاور منتقل می شوند. بنابراین دو ردیف شیفت می خورند تا به ردیف بالایی مربع مجاور برسند. ردیف بالایی مربع مجاور برسند. ردیف بالایی مربع مجاور یعنی ماتریس x آن مربع!

## گزارش ۲:

زمان		بسط داده شده	مشاهده شده	عمق	تست کیس			
۰.۷۵	)	0.141	200	۵	1			
۲۵۰		1419470.	۶۷۸۸۷۶۳	γ	٢			
Activities 🕒 Terr	minal		Fri Nov 17 22:37:14		📤 en 🔻 🛭 📢 년 51%			
F		mahd	-1-main	Q = - • ×				
SOLVING actions: [11, 7, 9, 1, 6], expanded nodes: 50141, explored nodes: 23955 actions: [11, 7, 9, 1, 6] SOLVE FINISHED In 0.75303s								
actions: [9, 7, 9, 7, 4, 3 expanded nodes: 14194750, explored nodes: 6788763 actions: [9, 7, 9 octions: SOLVE FINISHED In	9, 7, 4, 1, 6] n 248.01151s. ENTER TO VISUAL	python3 main.pytesto	case testcases/2.txtmethod	d IDS-DFS				

### گزارش ۳:

عدد خدا، به نتیجهای گفته می شود که نشان می دهد در هر تیب مسئله جست وجو حداکثر با چند حرکت می توان مسئله جست وجو است. جست وجو حل نشده را به حالت هدف برد. در واقع بالاترین عدد خدا برای هر مسئله هر مسئله جست وجو است. برای مکعب روبیک ۲×۲×۲ این عدد ۱۴ است. (با فرض اینکه هر حرکت فقط و فقط ۹۰ درجه است.)

## گزارش ۴:

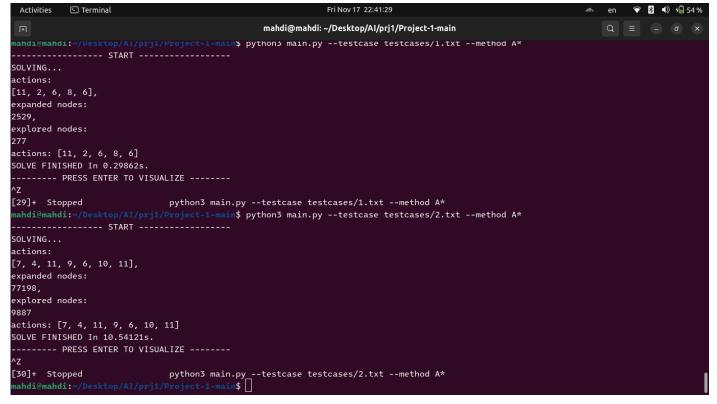
خیر موفق نشد که آن را حلکند. چون جواب در عمق بیشتری قرار دارد بنابراین DFS در هر مرحله قبل از لایه جواب کل درخت بالای آن لایه را گشته و تعداد گرههای بسط داده شده و سپس مشاهده شده آن به صورت نمایی افزایش می یابد. یعنی مثلا اگر حالت اولیه عمق 0 باشد 1 حالت داریم. در عمق 1، 1۲ تا در عمق ۲، ۱۴۴ و ... مثلا در عمق 5، 750 هزار گره باید محاسبه شود! با توجه به God's Number درست است که حالتی که تعداد حرکت بالایی می طلبد احتمالش کمتر است اما چون باز هم امکان آن وجود دارد حتی با فرض اینکه حداکثر عمق 15 داریم عدد در اوردر 1.۲ به توان 15 است!

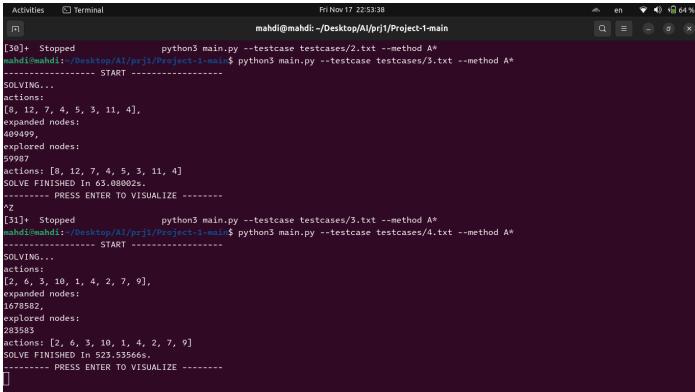
### گزارش ۵:

در واقع ما برای نشان دادن مکان هر مکعب کوچک از یک تنسور شامل دو ماتریس ۲ در ۲ استفاده کردیم. وقتی عملیات ۴ روی آن انجام شود یعنی لایه بالایی آن ۹۰ درجه میچرخد. چرخیدن لایه بالایی یعنی ۴ عنصر یا مکعب کوچک بالا چرخیده است. آدرس نشان دهنده چنین مکعبهایی در ماتریس ۱ و ۲ یعنی کل تنسور ردیف اول یعنی ایندکس و کل ستون موجود در آن است. بنابراین با ذخیره ردیف بالایی ماتریس ۱ و ۲ و دادن به تابع rot90 میتوان به آدرس جدید دست یافت. عدد ۱ نشان دهنده چرخش ساعت گرد است.

# گزارش ۶:

زمان جستوجو	بسط داده شده	مشاهده شده	عمق	تست کیس	نوع
۰.۷۵	۵۰۱۴۱	۲۳۹۵۵	۵	١	تعمیق تکراری
۲۵۰	14194100.	9711794	γ	٢	تعمیق تکراری
۰.۳ تا ۲.۰	7279	777	۵	١	A*
۱۰.۵	٧٧١٩٨	٩٨٨٧	٧	٢	A*
۶٠	4.9499	۵۹۹۸۷	٨	٣	A*
۵۲۰	184884	۲۸۳۵۸۳	٩	۴	A*



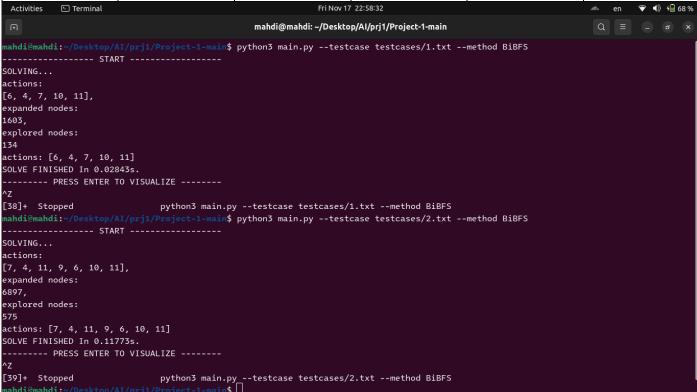


## گزارش ۷:

با توجه به در نظر گرفتن هیورستیک و مرتب کردن آنها بسیاری از گرهها به اولویت پاپ شدن نمیرسند به همین خاطر الگوریتم \*A هیچگاه نیاز ندارند گره های دور از مسیر هدف را را مشاهده کند در صورتی که IDS گرههای دور از هدف را نیز صرف اینکه در لایه بالایی قرار دارند ملزم به مشاهده کردن است همین امر سبب می شود تعداد گرههای مشاهده شده و بسط داده شده در \*A بسیار کمتر از IDS باشد.

## گزارش ۸:

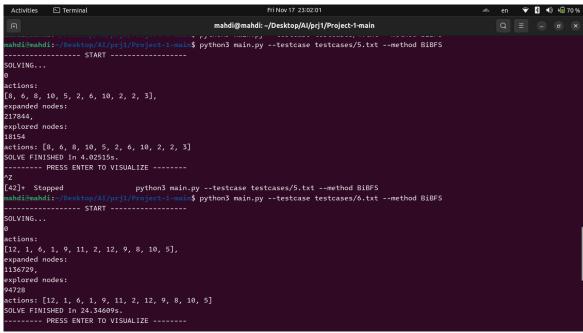
زمان	بسط داده شده	مشاهده شده	عمق	تست کیس	نوع
۰.۷۵	۵۰۱۴۱	۲۳۹۵۵	۵	١	تعمیق تکراری
۲۵۰	1419470.	977778	٧	٢	تعمیق تکراری
۲.۰ تا ۲.۰	7279	777	۵	١	A*
۱۰.۵	٨٩١٧٧	٩٨٨٧	٧	٢	A*
۶٠	4.9499	۵۹۹۸۷	٨	٣	A*
۵۲۰	184797	۲۸۳۵۸۳	٩	۴	A*
۰.۰۳ تا ۰.۰۲	18.4	174	۵	١	عرضی دو طرفه
۱.۰ تا ۱۲.۰	۶۸۹۷	۵۷۵	γ	۲	عرضی دو طرفه
۰.۵ تا ۰.۴	78940	7749	٨	٣	عرضی دو طرفه
۸.۰ تا ۱	۵۴۷۴۳	4087	٩	۴	عرضی دو طرفه
۴	717744	11124	11	۵	عرضی دو طرفه
74	1148749	94777	17	۶	عرضی دو طرفه
۲۷	17898.9	١٠۵٨٠١	١٣	γ	عرضی دو طرفه

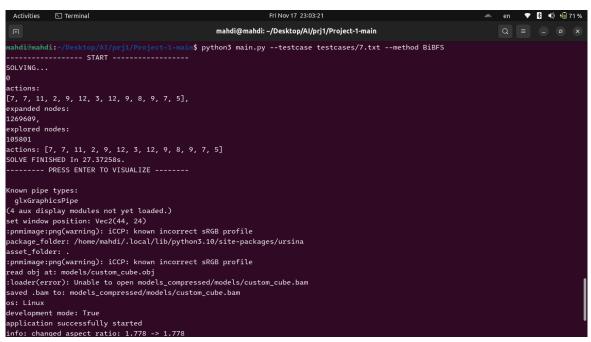


```
▼ 🔢 🜓 t🗎 68 9
                                                         mahdi@mahdi: ~/Desktop/AI/prj1/Project-1-main
                                                                                                                                      Q = -
 mahdi@mahdi:~/Desktop/AI/pxj1/Pxoject-1-main$ python3 main.py --testcase testcases/3.txt --method BiBFS
SOLVING...
actions:
[8, 12, 7, 4, 5, 3, 11, 4], expanded nodes:
26945,
explored nodes:
2246
actions: [8, 12, 7, 4, 5, 3, 11, 4]
SOLVE FINISHED In 0.45940s.
------ PRESS ENTER TO VISUALIZE ------
 40]+ Stopped python3 main.py --testcase testcases/3.txt --method BiBFS

ahdi@mahdi:~/Desktop/AI/pxj1/Project-1-main$ python3 main.py --testcase testcases/4.txt --method BiBFS

ON VING
actions:
[2, 6, 9, 9, 5, 9, 7, 2, 9], expanded nodes:
54743,
explored nodes:
4562
[41]+ Stopped
                                   python3 main.py --testcase testcases/4.txt --method BiBFS
                                                                       Fri Nov 17 23:02:01
                                                         mahdi@mahdi: ~/Desktop/AI/prj1/Project-1-main
                                                                                                                                      Q = - @
 ahdi@mahdi:~/Desktop/AI/prj1/Project-1-main$ python3 main.py --testcase testcases/5.txt --method BiBFS
  ----- START -----
SOLVING...
[8, 6, 8, 10, 5, 2, 6, 10, 2, 2, 3], expanded nodes:
```





#### گزارش ۹:

روش Bi-BFS به دلیل اینکه همزمان از سمت مبدا و هدف شروع به گسترش پیدا می کند؛ بسیار سریع تر است و تعداد گرههای کمتری نیز گسترش می یابد و مشاهده می شود.

روش \*A نیز درست است که کمتر از IDS طول میکشد اما همچنان چون امکان رسیدن به هیورستیک ایده آل تقریبا غیرممکن است و بار محاسباتی بالایی دارد بنابراین نمی تواند به خوبی Bi-BFS باشد. بنابراین تست کیسهای پیچیده تر حل آن غیر ممکن است.

روش IDS صرفا بدرد مسیرهای کوتاه سمت چپی درخت جستوجو میخورد بنابراین بسیار بسیار ضعیفتر از سایر روشها و با تعداد گره مشاهدهشده و بسط داده شده چندین برابری است.

## گزارش ۱۰:

الگوریتم با توجه به بار محاسباتی کمتر توانست تست کیسهای رندم را به خوبی حل کند. نمونههایی از آن در زیر آمده است.

```
Fri Nov 17 23:41:32
 Activities 🕒 Terminal
                                                                                                                 mahdi@mahdi: ~/Desktop/AI/prj1/Project-1-main
 ahdi@mahdi:~/Desktop/AI/prj1/Project-1-main$ python3 main.py --method BiBFS
 ----- START ------
SOLVING...
actions:
[4, 5, 4, 6, 10, 3, 5, 5, 8, 4],
expanded nodes:
194416,
explored nodes:
16202
actions: [4, 5, 4, 6, 10, 3, 5, 5, 8, 4]
SOLVE FINISHED In 3.51399s.
 ----- PRESS ENTER TO VISUALIZE -----
[6]+ Stopped
                           python3 main.py --method BiBFS
mahdi@mahdi:~/Desktop/AI/prj1/Project-1-main$ python3 main.py --method BiBFS
 ----- START ------
actions:
[10, 7, 10, 10, 8, 4, 6, 8, 10, 11, 8],
expanded nodes:
368008,
explored nodes:
30668
actions: [10, 7, 10, 10, 8, 4, 6, 8, 10, 11, 8]
SOLVE FINISHED In 7.05712s.
 ----- PRESS ENTER TO VISUALIZE -----
                            python3 main.py _-method BiBFS
```

```
Fri Nov 17 23:41:25
 Activities   Terminal
                                                                                                                    ♥ 🖁 🜓 🕫 93%
                                                                                                               Q = -
                                               mahdi@mahdi: ~/Desktop/AI/prj1/Project-1-main
mahdi@mahdi:~/Desktop/AI/prj1/Project-1-main$ python3 main.py --method BiBFS
 ----- START -----
SOLVING...
actions:
[6, 7, 5, 3, 5, 2, 7, 10, 8, 10, 6],
expanded nodes:
200642,
explored nodes:
16721
actions: [6, 7, 5, 3, 5, 2, 7, 10, 8, 10, 6]
SOLVE FINISHED In 3.62947s.
----- PRESS ENTER TO VISUALIZE -----
^Z
[4]+ Stopped
                           python3 main.py --method BiBFS
mahdi@mahdi:~/Desktop/AI/prj1/Project-1-main$ python3 main.py --method BiBFS
 ----- START ------
SOLVING...
actions:
[4, 2, 7, 4, 11, 3, 4, 9, 12, 10],
expanded nodes:
82079,
explored nodes:
6840
actions: [4, 2, 7, 4, 11, 3, 4, 9, 12, 10]
SOLVE FINISHED In 1.44489s.
----- PRESS ENTER TO VISUALIZE -----
[5]+ Stopped
                            python3 main.py --method BiBFS
mahdi@mahdi:~/Desktop/AI/prj1
                            /Project-1-main$ python3 main.py --method BiBFS
```

