«به نام خدا»

گزارش پروژه ۱ درس مبانی هوش مصنوعی

مهدی شاهینی ۹۹۲۳۰۴۰

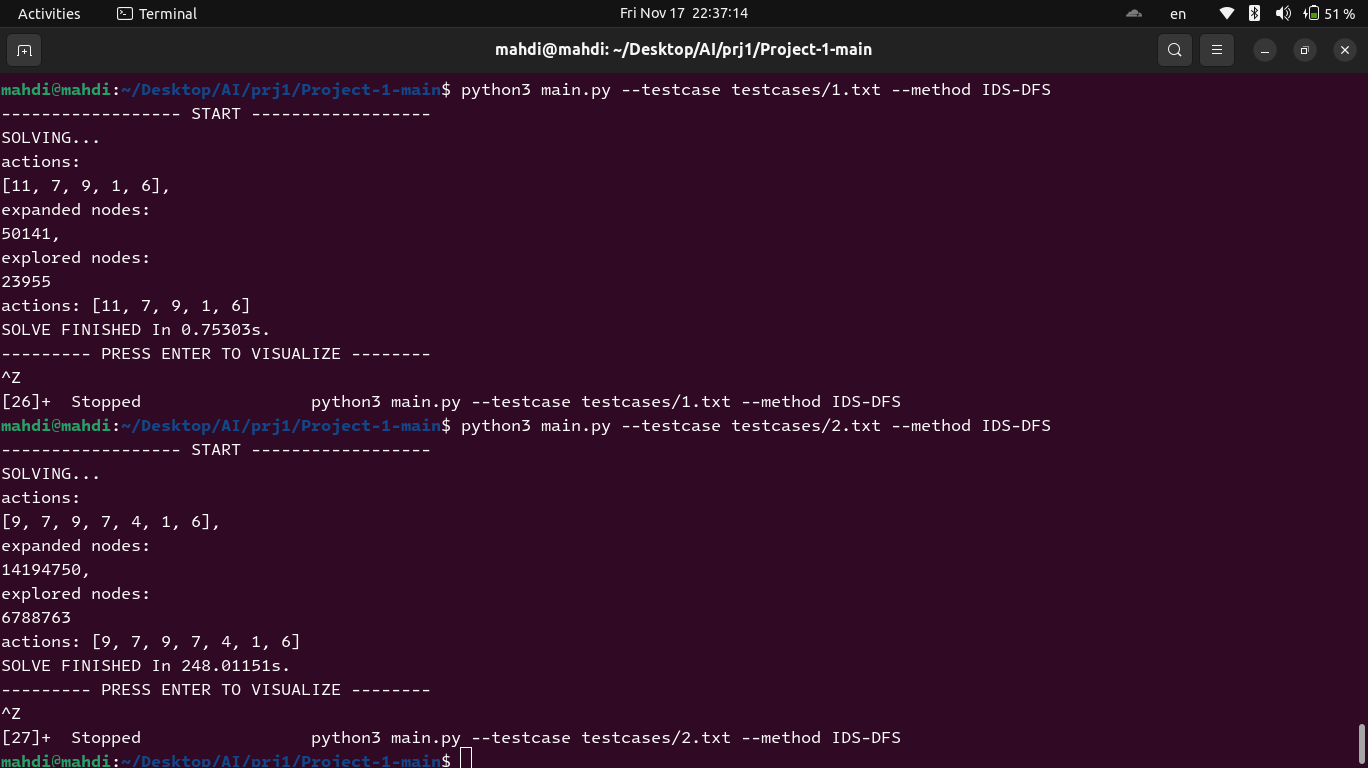
نکته: در الگوریتم‌های ۲ و ۳ زمانی که چندبار ران شد چند جواب مختلف در مدت زمان حدودی یکسان بدست آمد که همگی جواب صحیح بودند لطفا هنگام چک کردن صحت پاسخ‌ها گرافیک پاسخ را نیز مشاهده کنید.

گزارش ۱:

فعالیت ۴، در واقع چرخش ۹۰ درجه ساعت گرد صفحه بالایی روبیک است. بنابراین باید ماتریسی که به این ناحیه اشاره می‌کند ۹۰ درجه بچرخد. دو ردیف اول ماتریس state به این صفحه اشاره می‌کند که ۲ ردیف اول با دستور rot90 می‌چرخد حال باید خانه‌های جانبی که تغییر کرده‌اند را بروزرسانی کنیم. برای این کار چون هر دو ردیف یک مربع را نشان می‌دهند در واقع با چرخ ۹۰ درجه هر ردیف دو ردیف شیفت می‌خورد 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 می‌شود. علت این شیفت خوردن این است که در واقع از هر مربع به مربع مجاور منتقل می‌شوند. بنابراین دو ردیف شیفت می‌خورند تا به ردیف بالایی مربع مجاور برسند. ردیف بالایی مربع مجاور یعنی ماتریس ۲×۲ آن مربع!

**گزارش ۲:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| تست کیس | عمق | مشاهده شده | بسط داده شده | زمان |
| ۱ | ۵ | ۲۳۹۵۵ | ۵۰۱۴۱ | ۰.۷۵ |
| ۲ | ۷ | ۶۷۸۸۷۶۳ | ۱۴۱۹۴۷۵۰ | ۲۵۰ |



گزارش ۳:

عدد خدا، به نتیجه‌ای گفته می‌شود که نشان می‌دهد در هر تیب مسئله جست‌وجو حداکثر با چند حرکت می‌توان مسئله جست‌وجو حل نشده را به حالت هدف برد. در واقع بالاترین عدد خدا برای هر مسئله worst case آن مسئله جست‌وجو است.

برای مکعب روبیک ۲×۲×۲ این عدد ۱۴ است. (با فرض اینکه هر حرکت فقط و فقط ۹۰ درجه است.)

گزارش ۴:

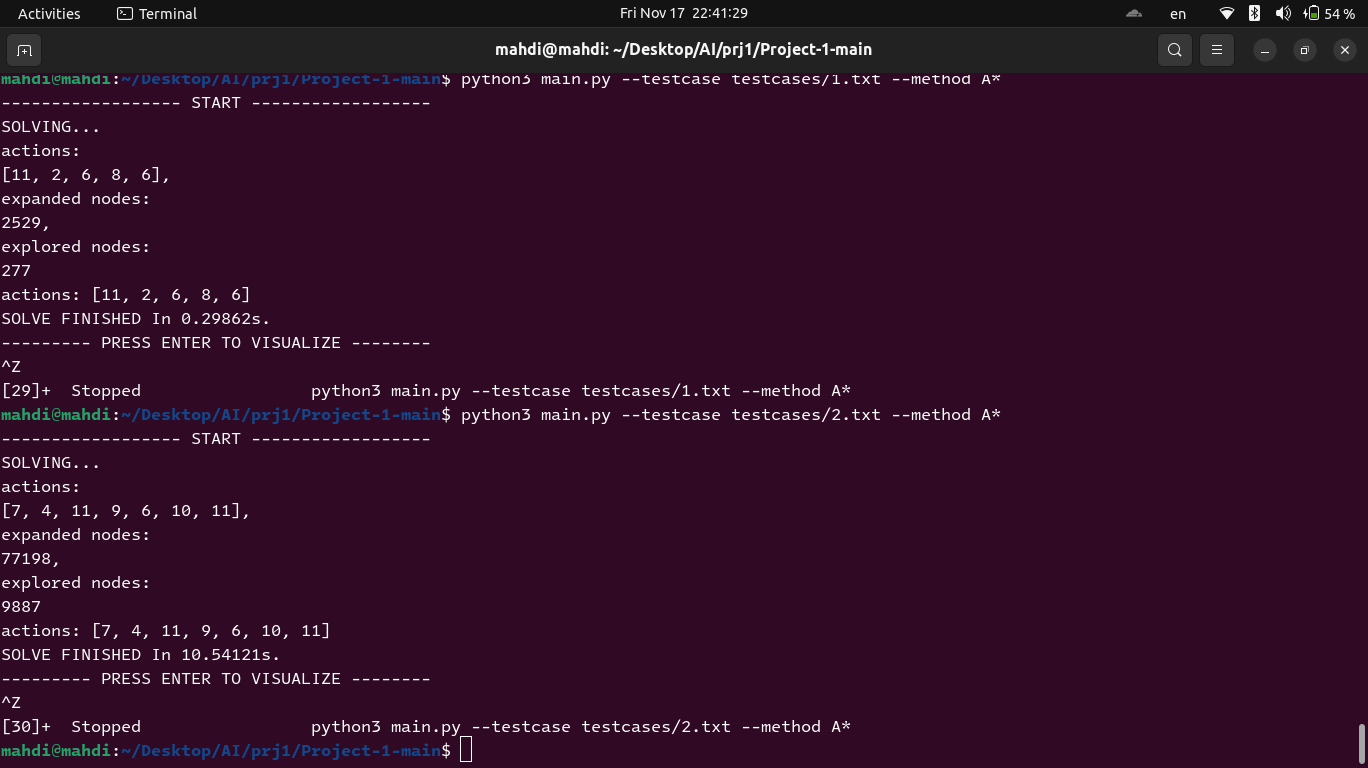
خیر موفق نشد که آن را حل‌کند. چون جواب در عمق بیش‌تری قرار دارد بنابراین DFS در هر مرحله قبل از لایه جواب کل درخت بالای آن لایه را گشته و تعداد گره‌های بسط داده شده و سپس مشاهده شده آن به صورت نمایی افزایش می‌یابد. یعنی مثلا اگر حالت اولیه عمق 0 باشد 1 حالت داریم. در عمق 1، 1۲ تا در عمق ۲، 1۴۴ و ... مثلا در عمق 5، ۲50 هزار گره باید محاسبه شود! با توجه به God’s Number درست است که حالتی که تعداد حرکت بالایی می‌طلبد احتمالش کمتر است اما چون باز هم امکان آن وجود دارد حتی با فرض اینکه حداکثر عمق 1۴ داریم عدد در اوردر 1.۲ به توان 15 است!

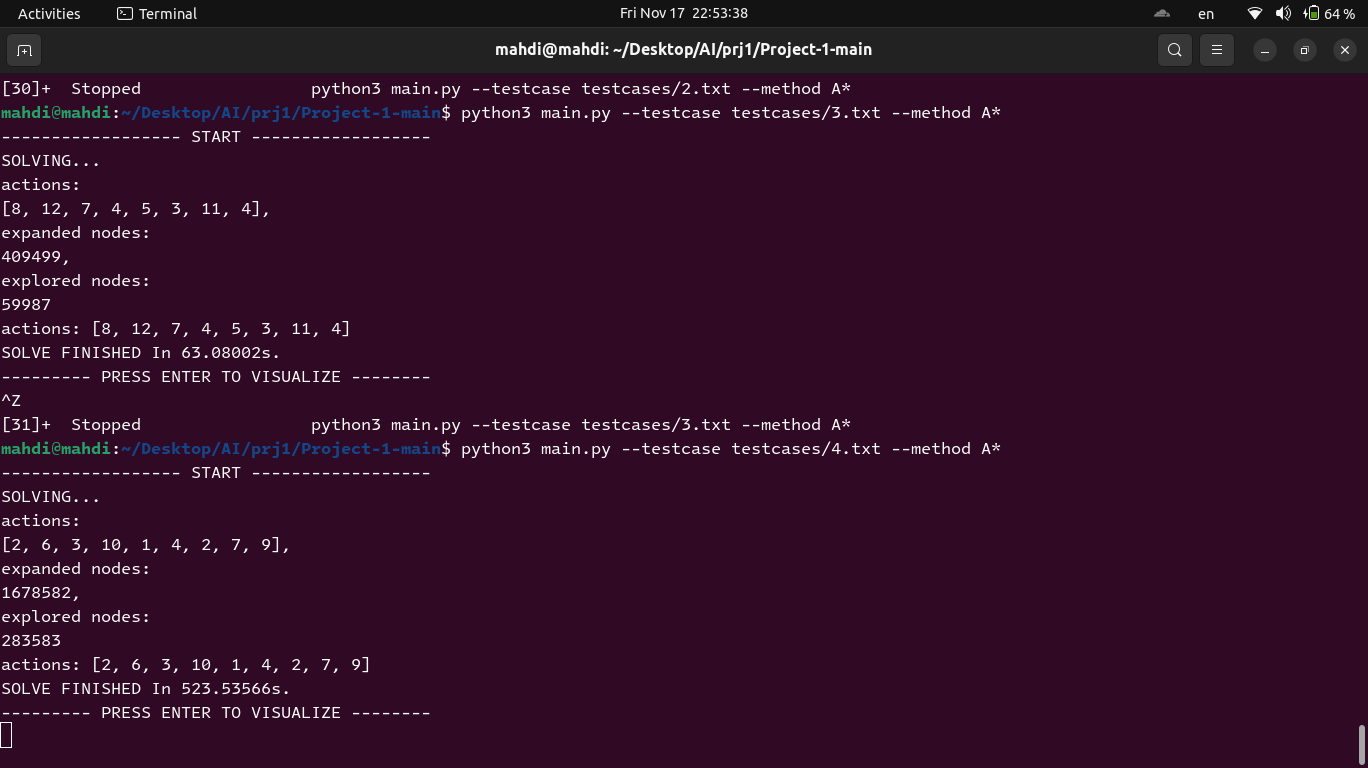
گزارش ۵:

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 ۰ 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:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| نوع | تست کیس | عمق | مشاهده شده | بسط داده شده | زمان جست‌و‌جو |
| تعمیق تکراری | ۱ | ۵ | ۲۳۹۵۵ | ۵۰۱۴۱ | ۰.۷۵ |
| تعمیق تکراری | ۲ | ۷ | ۶۷۸۸۷۶۳ | ۱۴۱۹۴۷۵۰ | ۲۵۰ |
| A\* | ۱ | ۵ | ۲۷۷ | ۲۵۲۹ | ۰.۲ تا ۰.۳ |
| A\* | ۲ | ۷ | ۹۸۸۷ | ۷۷۱۹۸ | ۱۰.۵ |
| A\* | ۳ | ۸ | ۵۹۹۸۷ | ۴۰۹۴۹۹ | ۶۰ |
| A\* | ۴ | ۹ | ۲۸۳۵۸۳ | ۱۶۷۸۵۸۲ | ۵۲۰ |



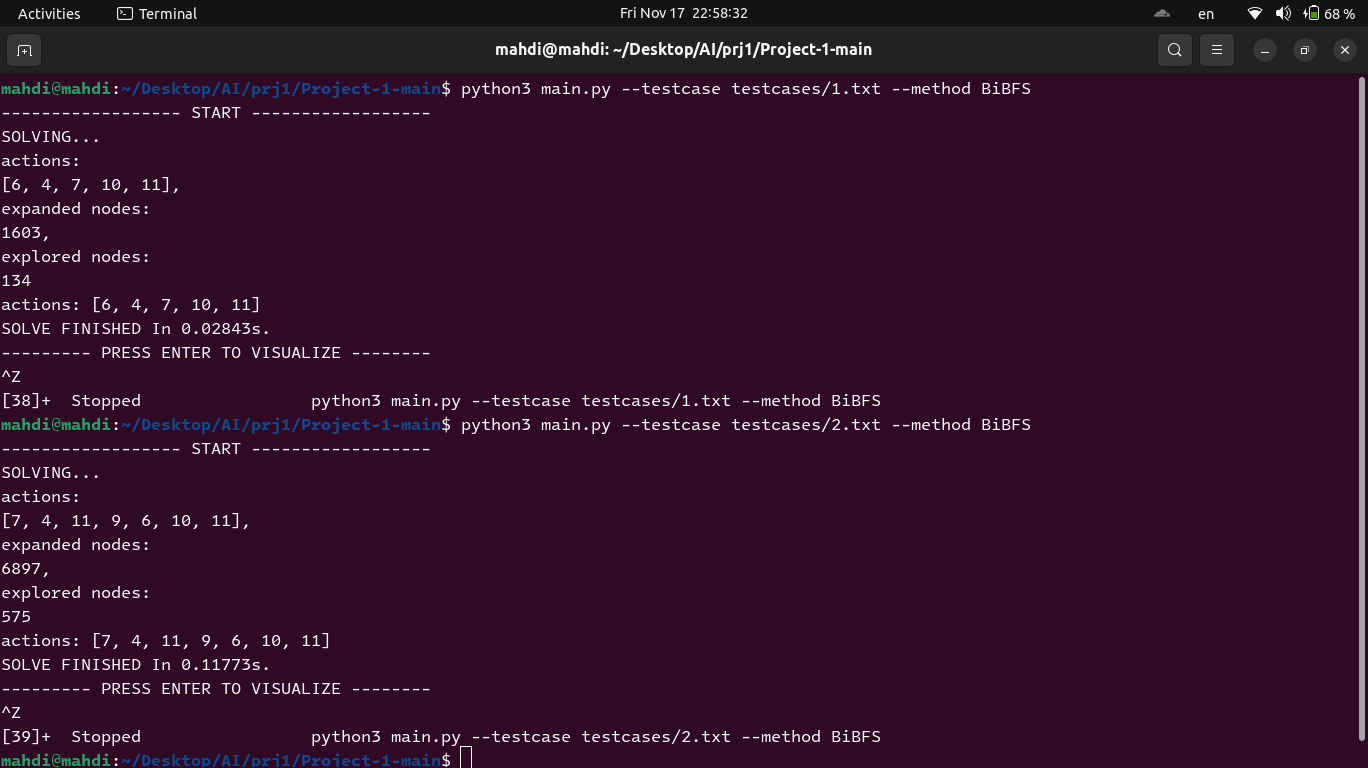


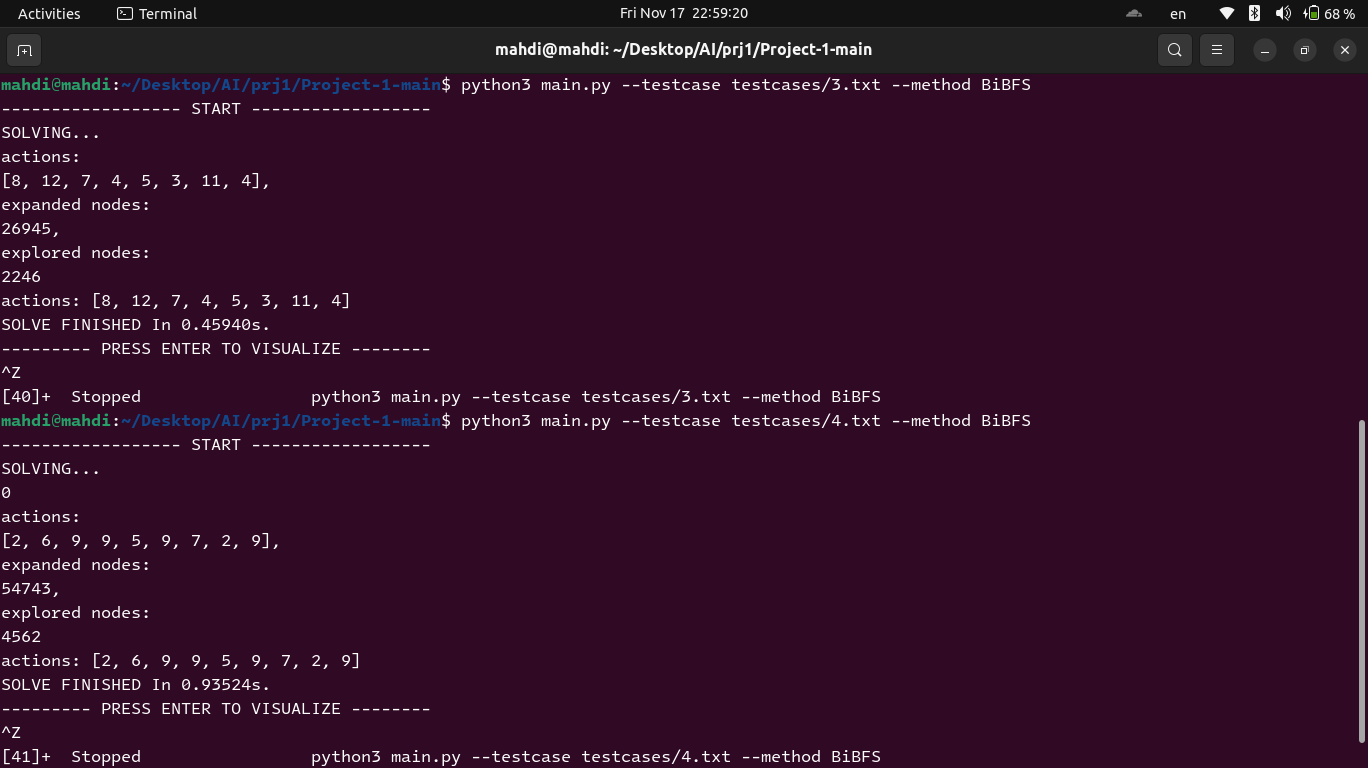
**گزارش ۷:**

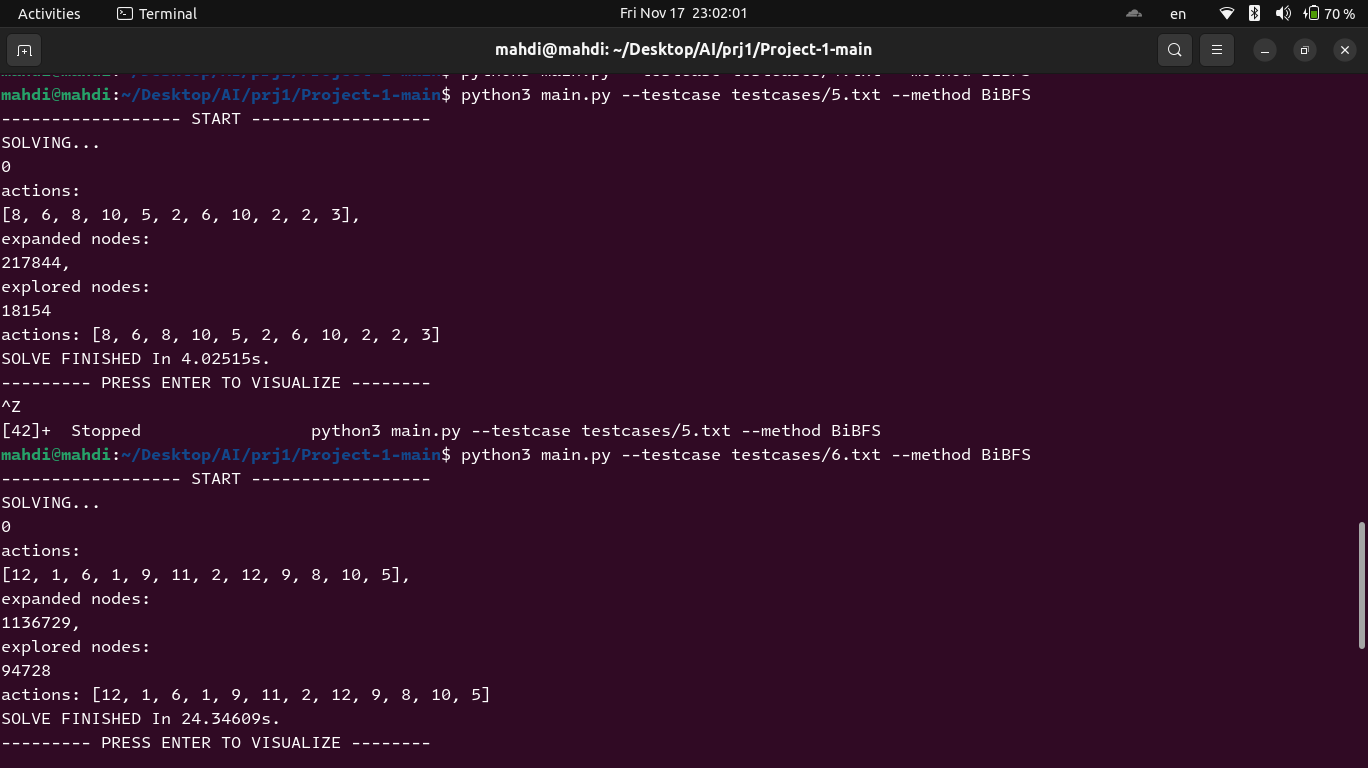
با توجه به در نظر گرفتن هیورستیک و مرتب کردن آن‌ها بسیاری از گره‌ها به اولویت پاپ شدن نمی‌رسند به همین خاطر الگوریتم A\* هیچ‌گاه نیاز ندارند گره های دور از مسیر هدف را را مشاهده کند در صورتی که IDS گره‌های دور از هدف را نیز صرف اینکه در لایه بالایی قرار دارند ملزم به مشاهده کردن است همین امر سبب می‌شود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

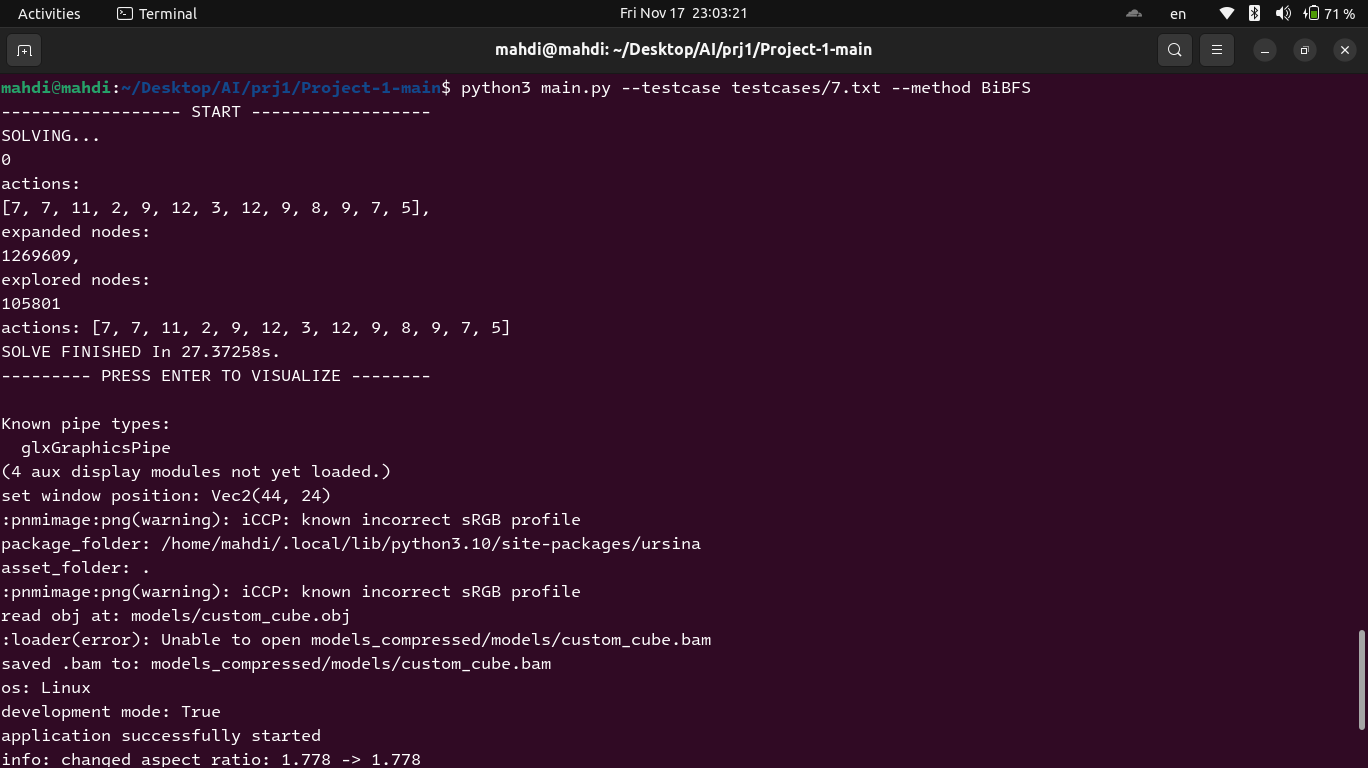
**گزارش ۸:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| نوع | تست کیس | عمق | مشاهده شده | بسط داده شده | زمان |
| تعمیق تکراری | ۱ | ۵ | ۲۳۹۵۵ | ۵۰۱۴۱ | ۰.۷۵ |
| تعمیق تکراری | ۲ | ۷ | ۶۷۸۸۷۶۳ | ۱۴۱۹۴۷۵۰ | ۲۵۰ |
| A\* | ۱ | ۵ | ۲۷۷ | ۲۵۲۹ | ۰.۲ تا ۰.۳ |
| A\* | ۲ | ۷ | ۹۸۸۷ | ۷۷۱۹۸ | ۱۰.۵ |
| A\* | ۳ | ۸ | ۵۹۹۸۷ | ۴۰۹۴۹۹ | ۶۰ |
| A\* | ۴ | ۹ | ۲۸۳۵۸۳ | ۱۶۷۸۵۸۲ | ۵۲۰ |
| عرضی دو طرفه | ۱ | ۵ | ۱۳۴ | ۱۶۰۳ | ۰.۰۲ تا ۰.۰۳ |
| عرضی دو طرفه | ۲ | ۷ | ۵۷۵ | ۶۸۹۷ | ۰.۱ تا ۰.۱۲ |
| عرضی دو طرفه | ۳ | ۸ | ۲۲۴۶ | ۲۶۹۴۵ | ۰.۴ تا ۰.۵ |
| عرضی دو طرفه | ۴ | ۹ | ۴۵۶۲ | ۵۴۷۴۳ | ۰.۸ تا ۱ |
| عرضی دو طرفه | ۵ | ۱۱ | ۱۸۱۵۴ | ۲۱۷۸۴۴ | ۴ |
| عرضی دو طرفه | ۶ | ۱۲ | ۹۴۷۲۸ | ۱۱۳۶۷۲۹ | ۲۴ |
| عرضی دو طرفه | ۷ | ۱۳ | ۱۰۵۸۰۱ | ۱۲۶۹۶۰۹ | ۲۷ |









گزارش ۹:

روش Bi-BFS به دلیل اینکه همزمان از سمت مبدا و هدف شروع به گسترش پیدا می‌کند؛ بسیار سریع‌تر است و تعداد گره‌های کمتری نیز گسترش می‌یابد و مشاهده می‌شود.

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 نمی‌تواند به خوبی Bi-BFS‌ باشد. بنابراین تست کیس‌های پیچیده‌تر 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.

