**گزارش پروژه هوش مصنوعی**

**۱)**



state[0:2, :] = np.rot90(state[0:2, :], -1)

در این خط ابتدا به وسیله تابع کتابخانه نامپای دو قسمت اول state که در واقع مربع های سطح بالایی میشوند را نود درجه ساعتگرد میچرخاند.

        start = np.copy(state[2, :])

در این خط دو مربع نصفه ی بالایی وجه سمت چپ را در یک متغیر نگه میدارد تا بتواند در آخر کار آنها را بر روی وجه پشتی قرار بدهد و از دست نرود.

        state[2, :] = state[4, :]

در این خط دو مربع نصفه ی بالایی وجه روبرو را روی دو مربع نصفه ی بالایی وجه چپ قرار میدهد.

        state[4, :] = state[6, :]

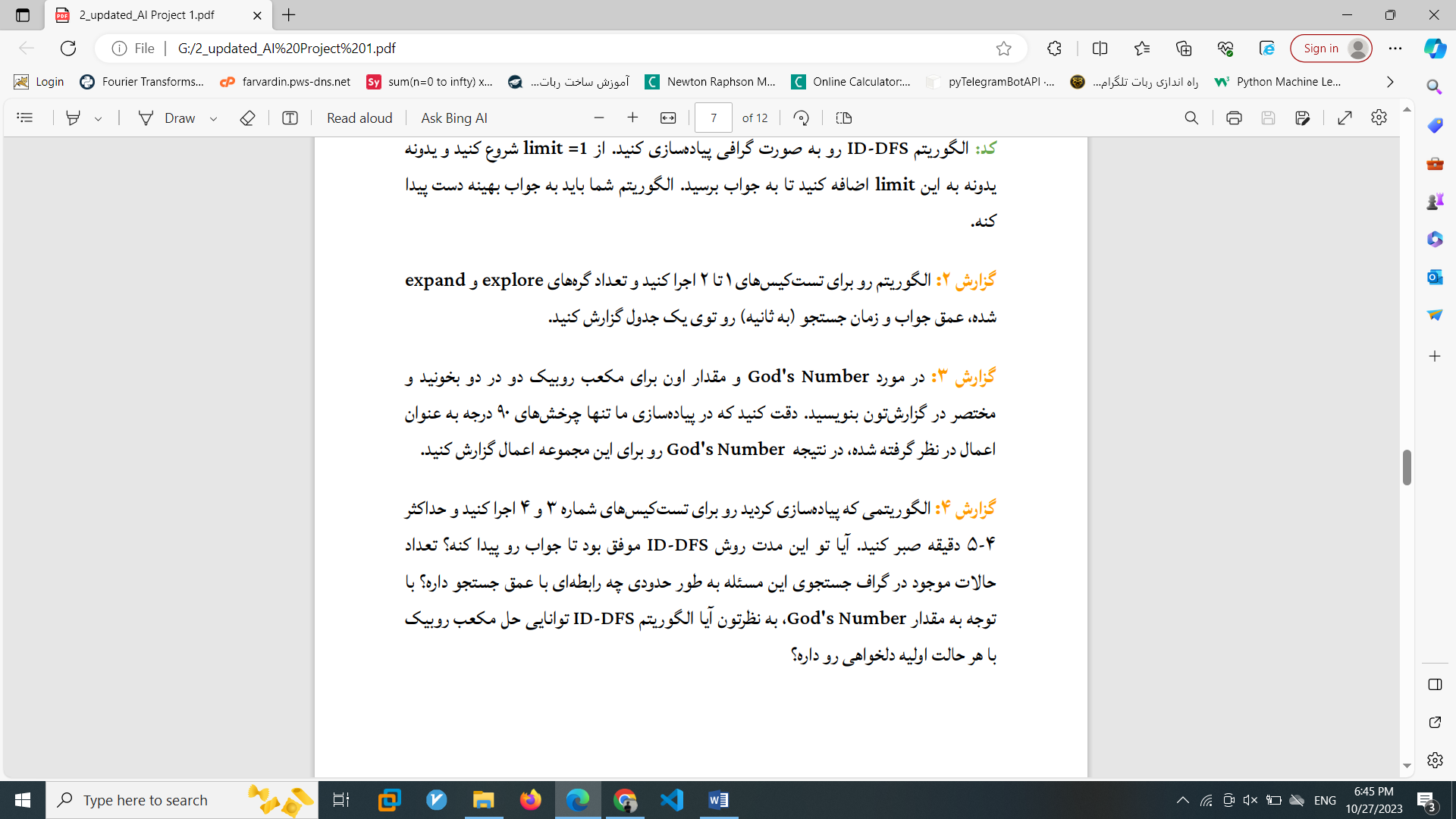
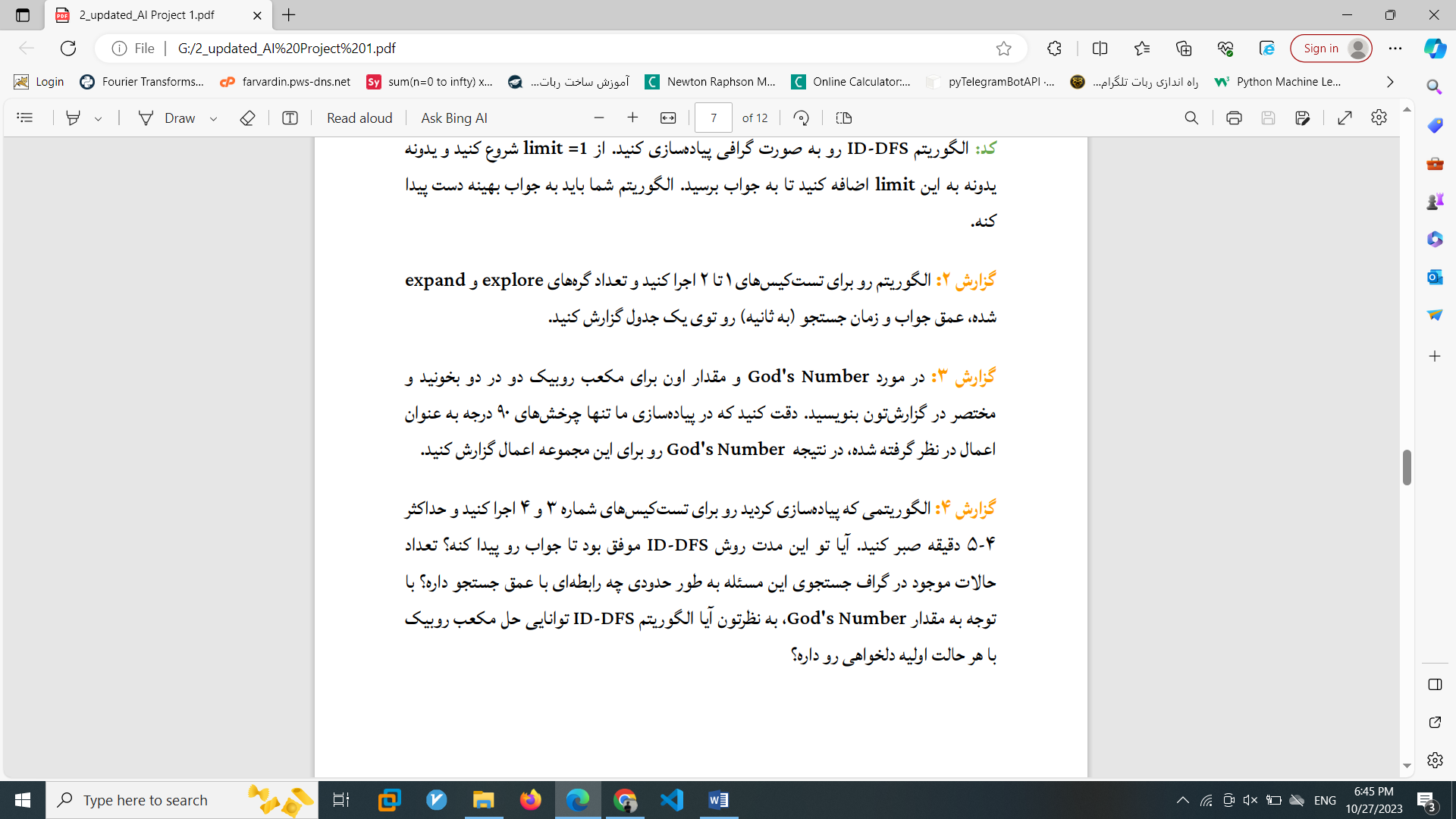
در این خط دو مربع نصفه ی بالایی وجه راست را روی دو مربع نصفه ی بالایی وجه روبرو قرار میدهد.

        state[6, :] = state[8, :]

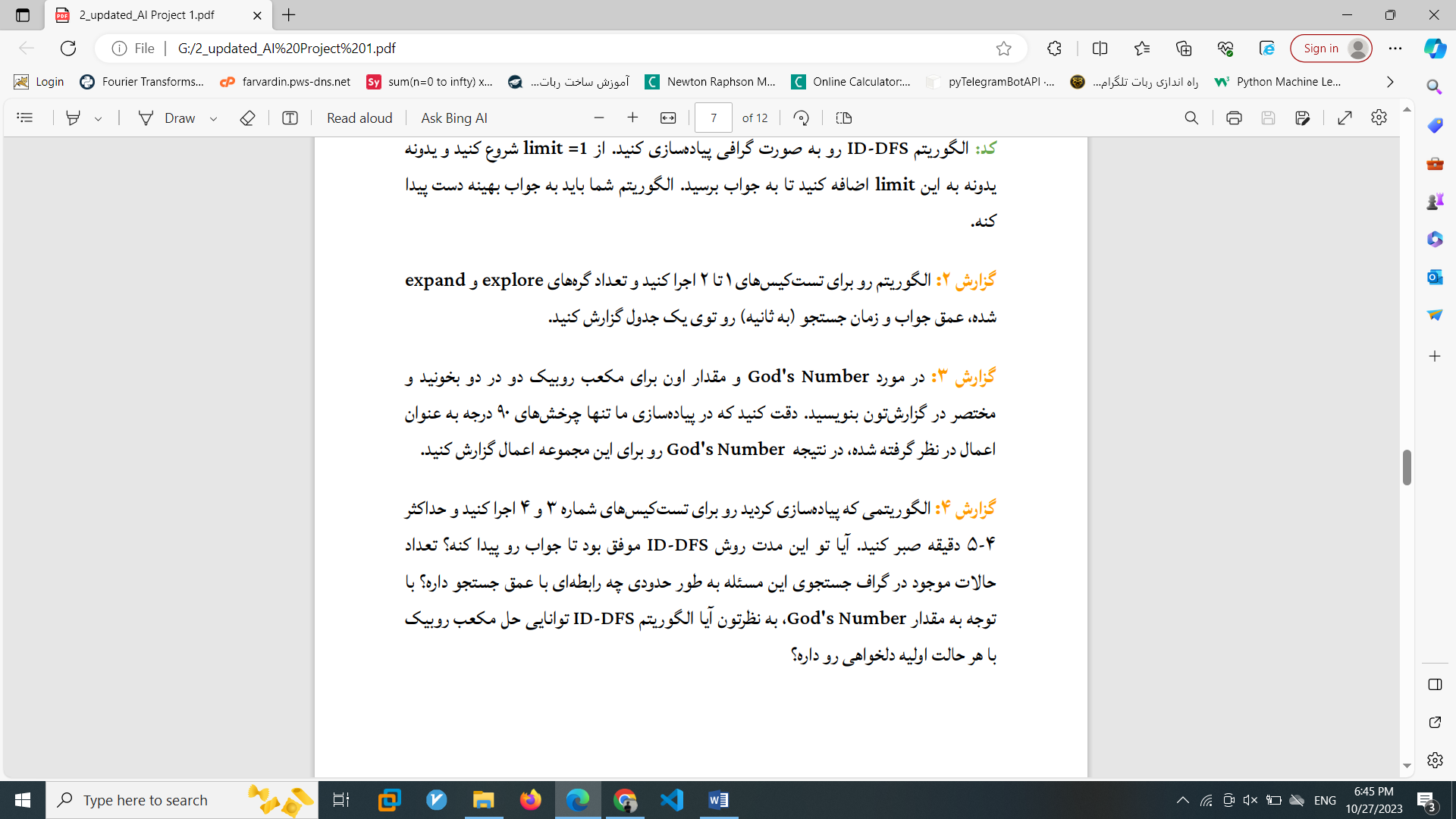
در این خط دو مربع نصفه ی بالایی وجه پشت را روی دو مربع نصفه ی بالایی وجه راست قرار میدهد.

        state[8, :] = start

در این خط دو مربع نصفه ی بالایی وجه چپ را که قبلا در یک متغیر دیگر ذخیره کرده است روی دو مربع نصفه ی بالایی وجه پشتی قرار میدهد.



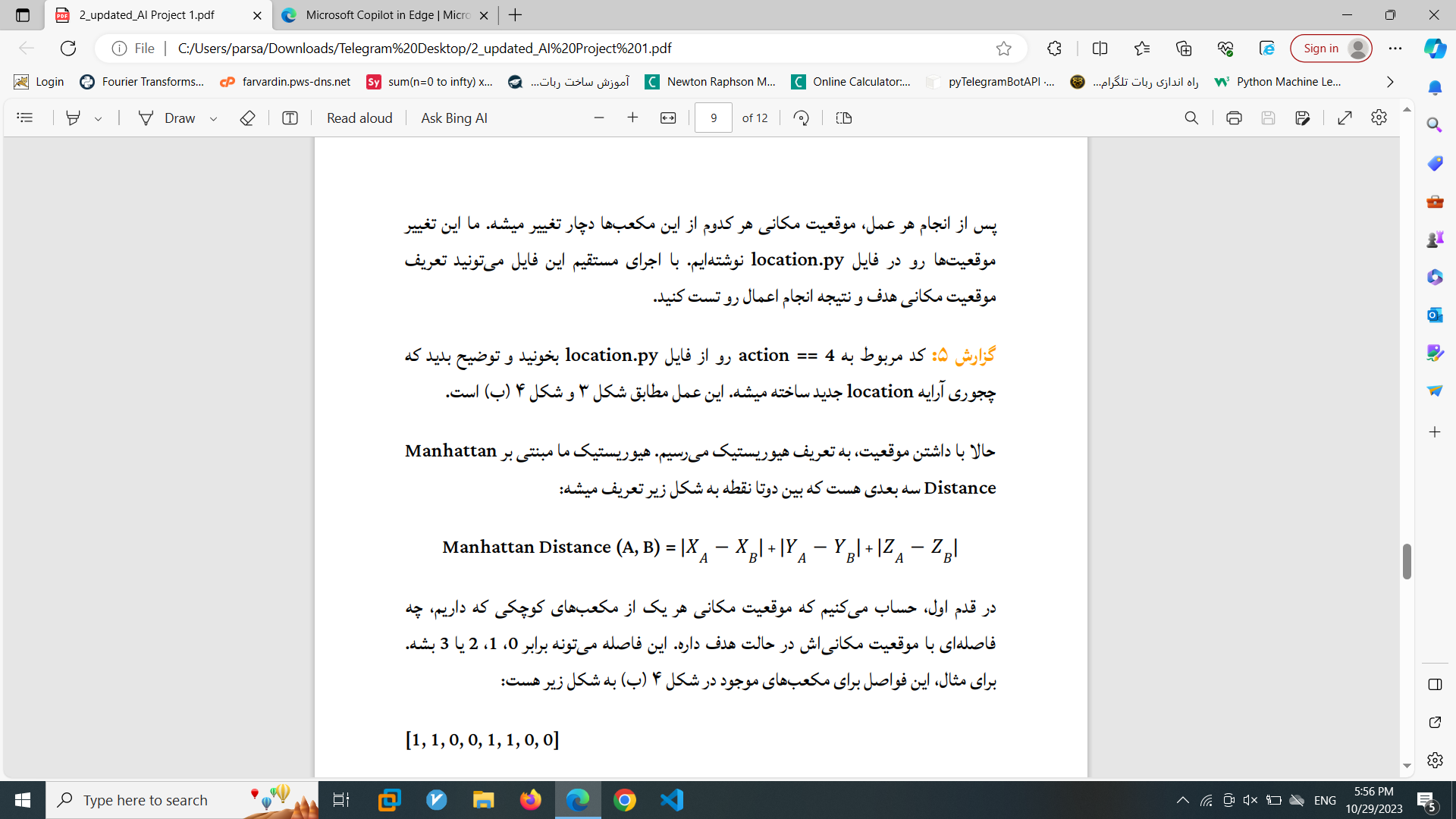
به حداقل تعداد حرکات لازم برای حل یک مکعب روبیک که به بدترین شکل خراب شده است god number میگویند که برای مکعب های دو در دو ۱۴ میباشد.



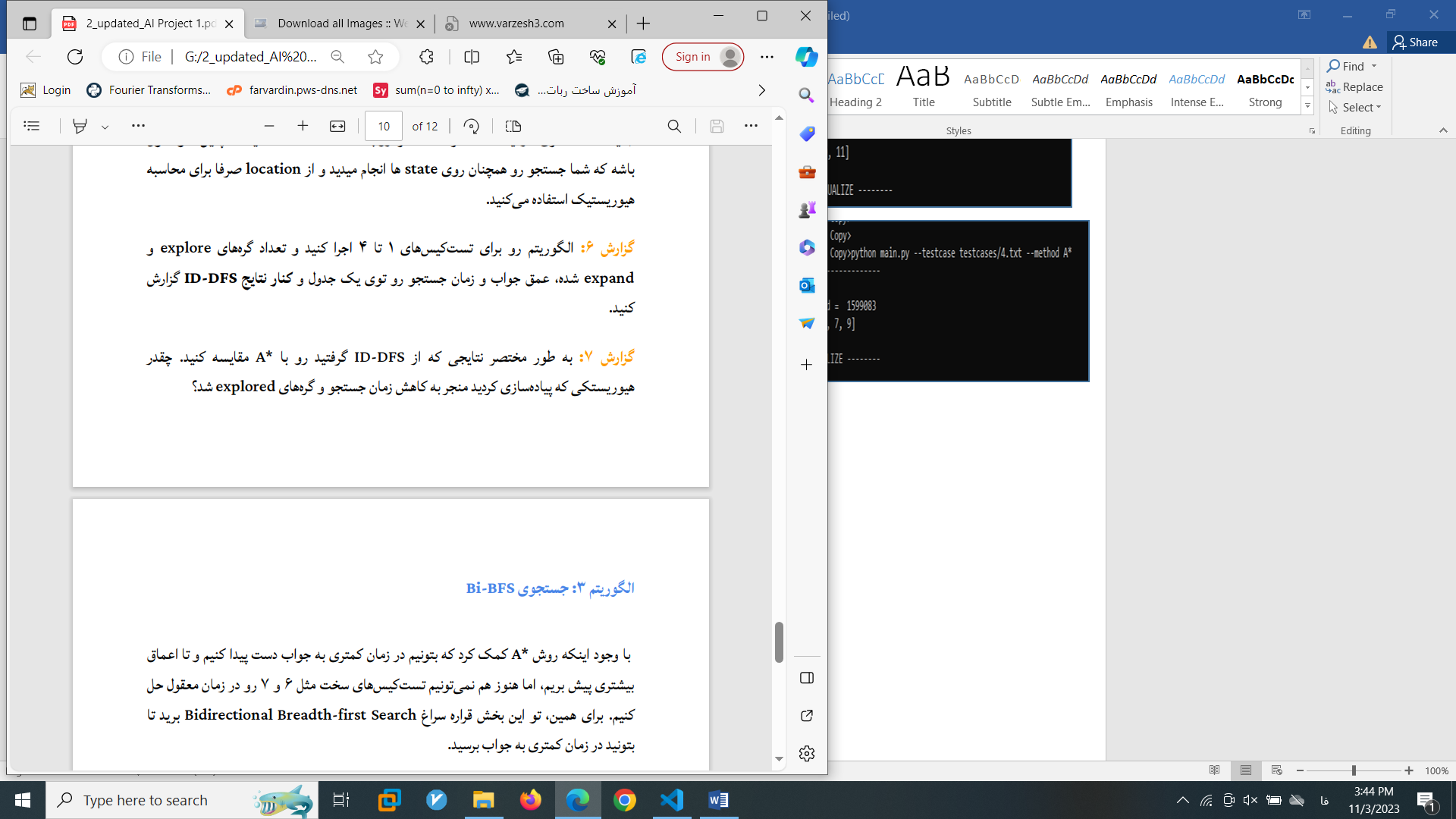
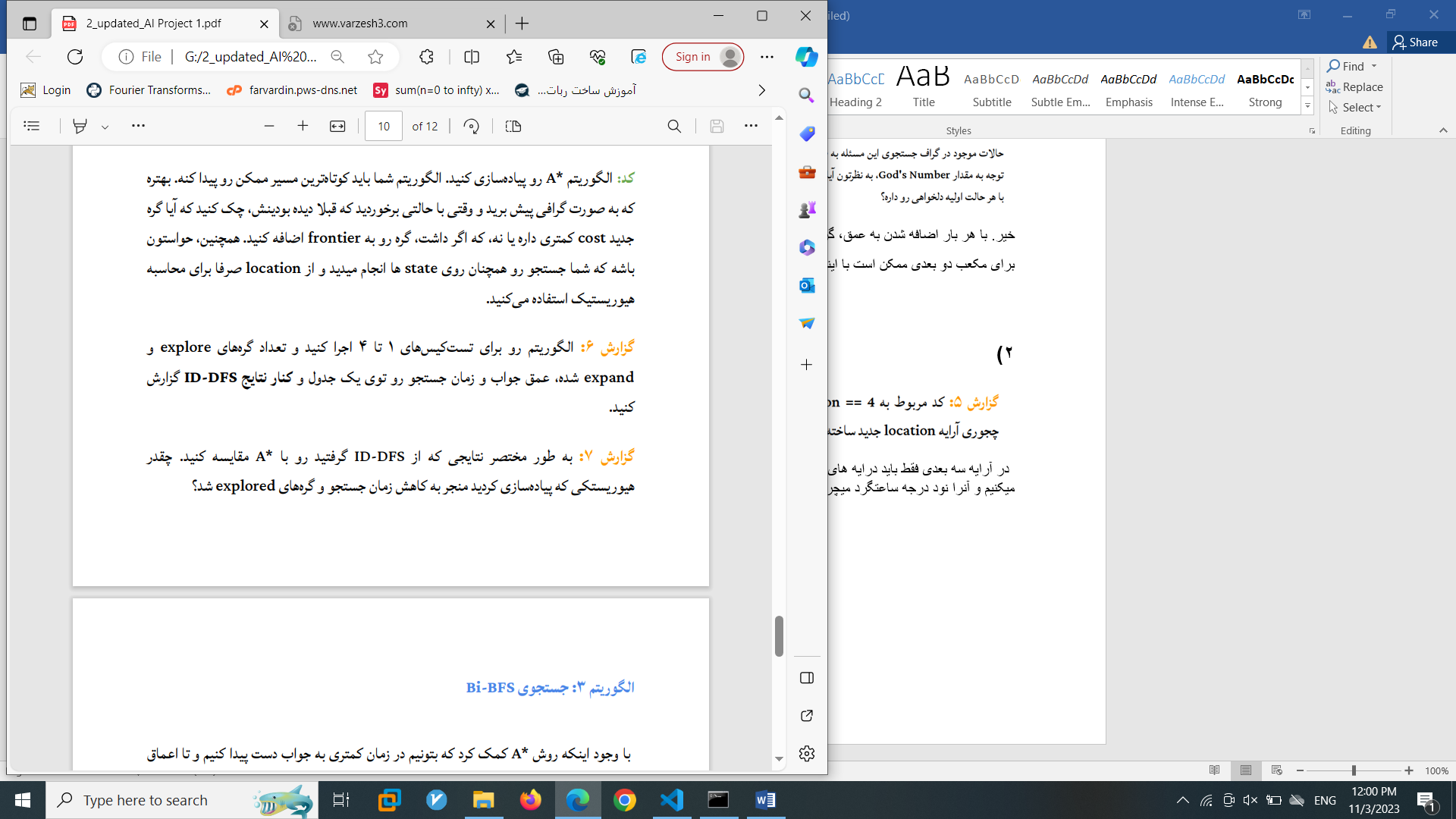
خیر. با هر بار اضافه شدن به عمق، گراف جستجو ۱۲ برابر میشود و به صورت نمایی زیاد میشود.

برای مکعب دو بعدی ممکن است با اینکه به زمان زیادی نیاز دارد ولی به طور کلی و در مدت کم امکان پذیر نیست.

**۲)**

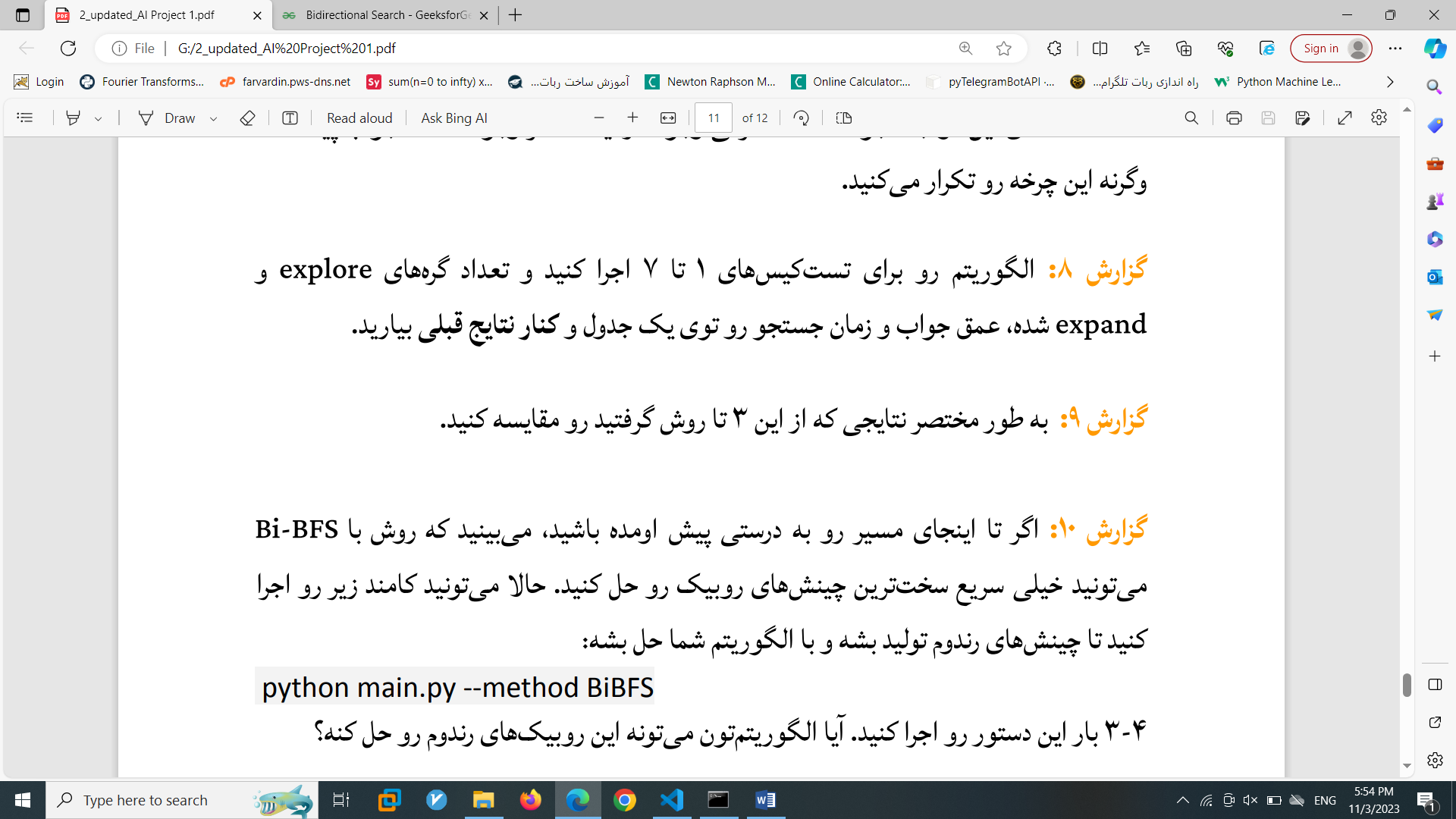


در آرایه سه بعدی فقط باید درایه های بخش بالایی جابجا شوند بنابراین در بعد دوم آرایه فقط ایندکس صفر را عوض میکنیم و آنرا نود درجه ساعتگرد میچرخانیم.



هم از نظر زمانی و هم از نظر تعداد گره های جستجو شده به مقدار بسیار زیاد و قابل توجه ای کم شده است.

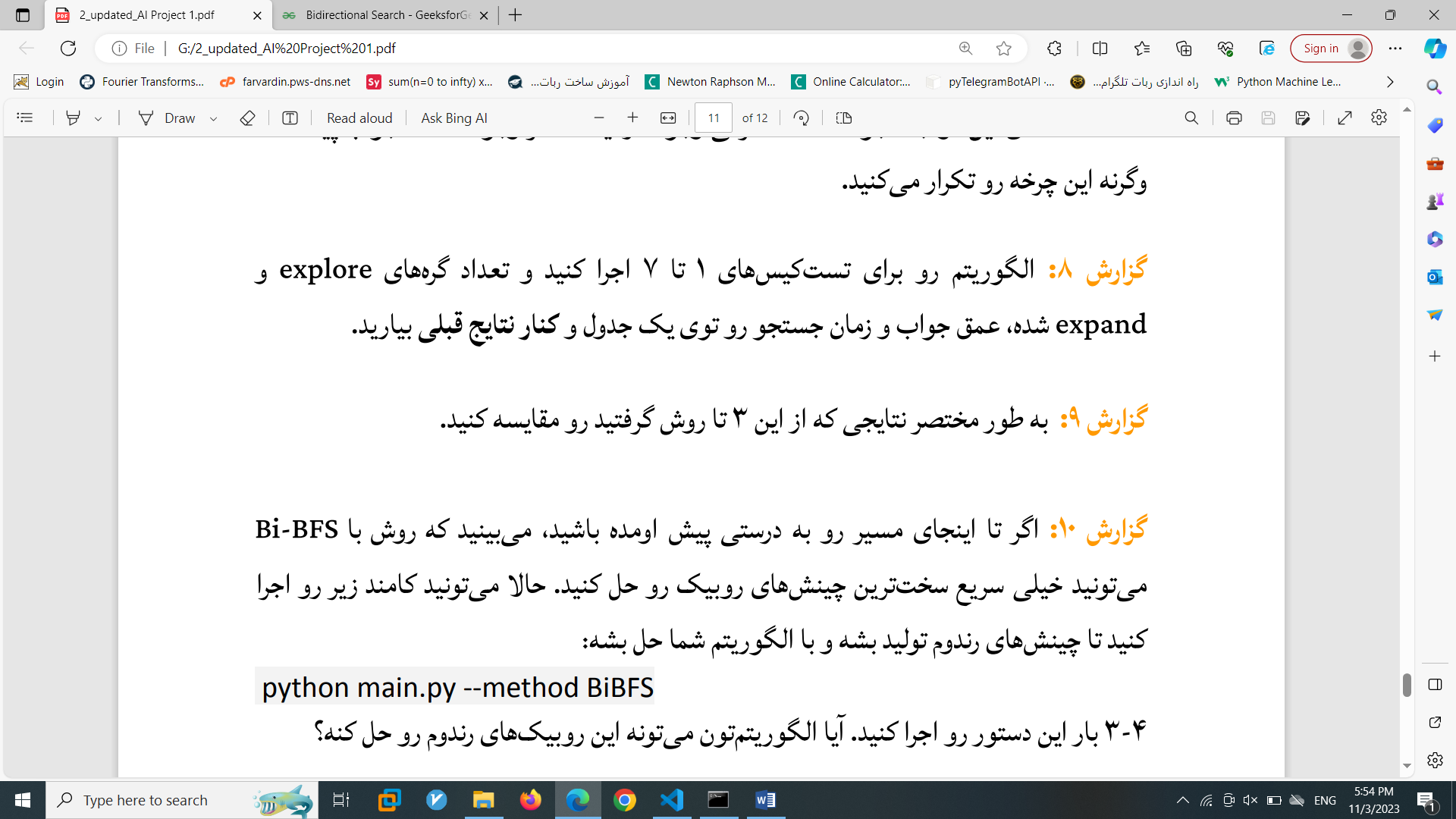
**۳)**



|  |  |  |
| --- | --- | --- |
| 2 | 1 | Id dfs |
| 14761872 | 57372 | explore |
| 14761872 | 57349 | expand |
| 444 | 1 | time |

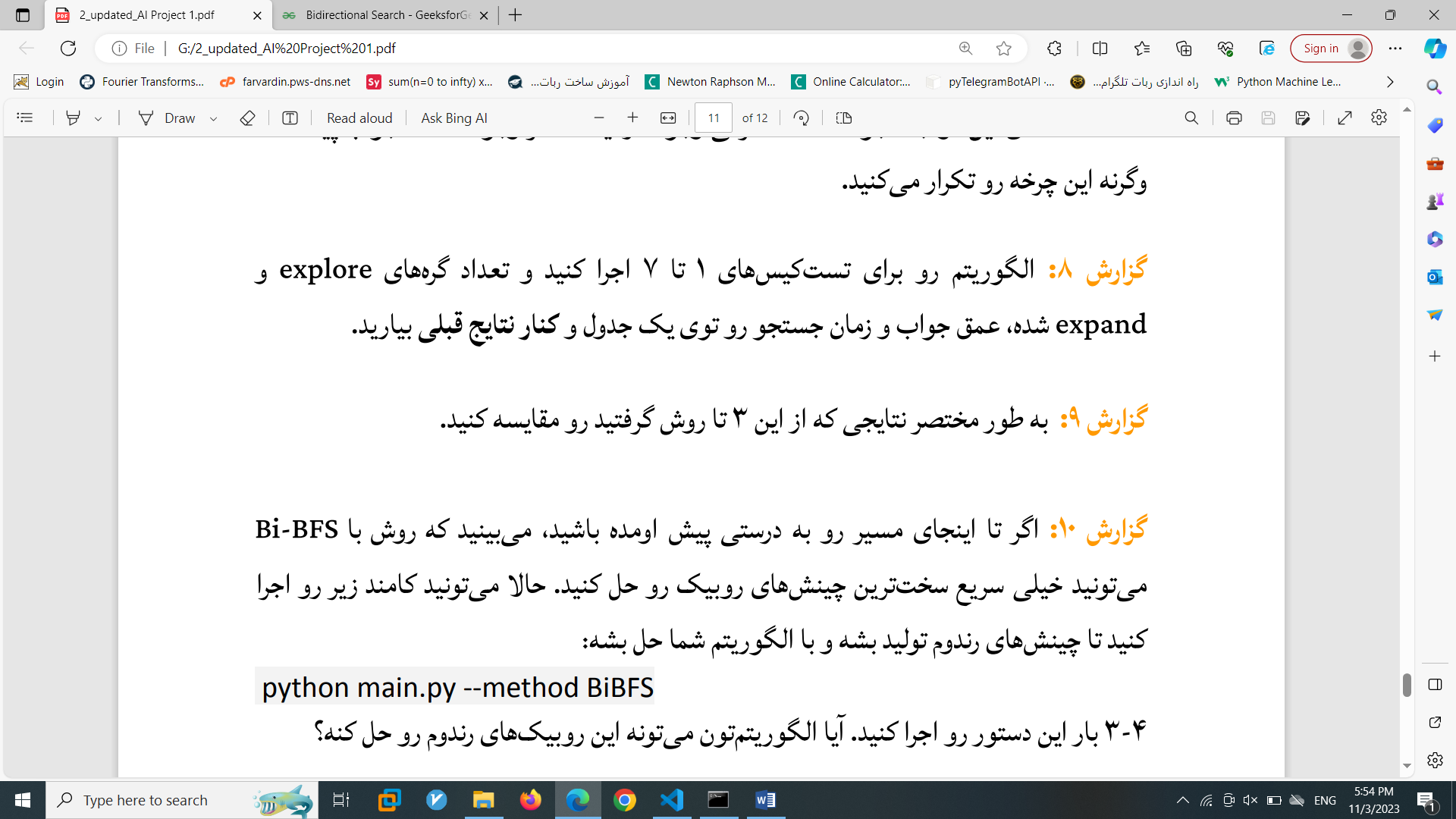
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | 3 | 2 | 1 | A\* |
| 1677971 | 409361 | 77185 | 2528 | explore |
| 283605 | 59990 | 9887 | 277 | expand |
| 231 | 102 | 9 | 0.26 | time |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | BiBFS |
| 183004 | 106851 | 25723 | 4805 | 1945 | 501 | 90 | explore |
| 1135830 | 679623 | 186432 | 39569 | 16588 | 4842 | 991 | expand |
| 84 | 150 | 8.6 | 1.3 | 0.75 | 0.27 | 0.1 | time |



در حالت کلی در روش bi bfs کاهش بسیار چشم گیری مشاهده میشود و خیلی سریعتر به جواب میرسیم

در حالی که در روش A\* نسبت به ةن ضعیفتر است ولی از id dfs بهتر است.



بله