حسن کاظمی طهرانی

گزارش پروژه 2

9629041

مبانی هوش

فرموله بندی مسئله به صورت زیر انجام شده است.

متغییر ها : رنگ هر یک از خانه های جدول و عدد هر یک از خانه های جدول

Ci,j:رنگ خانه i , j

Ni,j : شماره خانه i , j

دامنه ها: اگر جدول n\*n باشد :

Dn={1,2,3,..,n}

Dc={input of user}

محدودیت ها :

1- شماره هر خانه باید با شماره های خانه های موجود در سطر و ستون یکسان متفاوت باشد.

Ni,j  != Nk,m if (i==k) or (j==m)

2- رنگ هر خانه باید با خانه های مجاور متفاوت باشد.

Ci,j != Ci-1,j

Ci,j != Ci+1,j

Ci,j != Ci,j-1

Ci,j != Ci,j+

3- اگر عدد خانه های مجاور بزگتر بود اولویت رنگ نیز بزرگتر باشد.

If Ni-1,j > Ni,j => priority(Ci-1,j) > priority(Ci,j)

If Ni+1,j > Ni,j => priority(Ci+1,j) > priority(Ci,j)

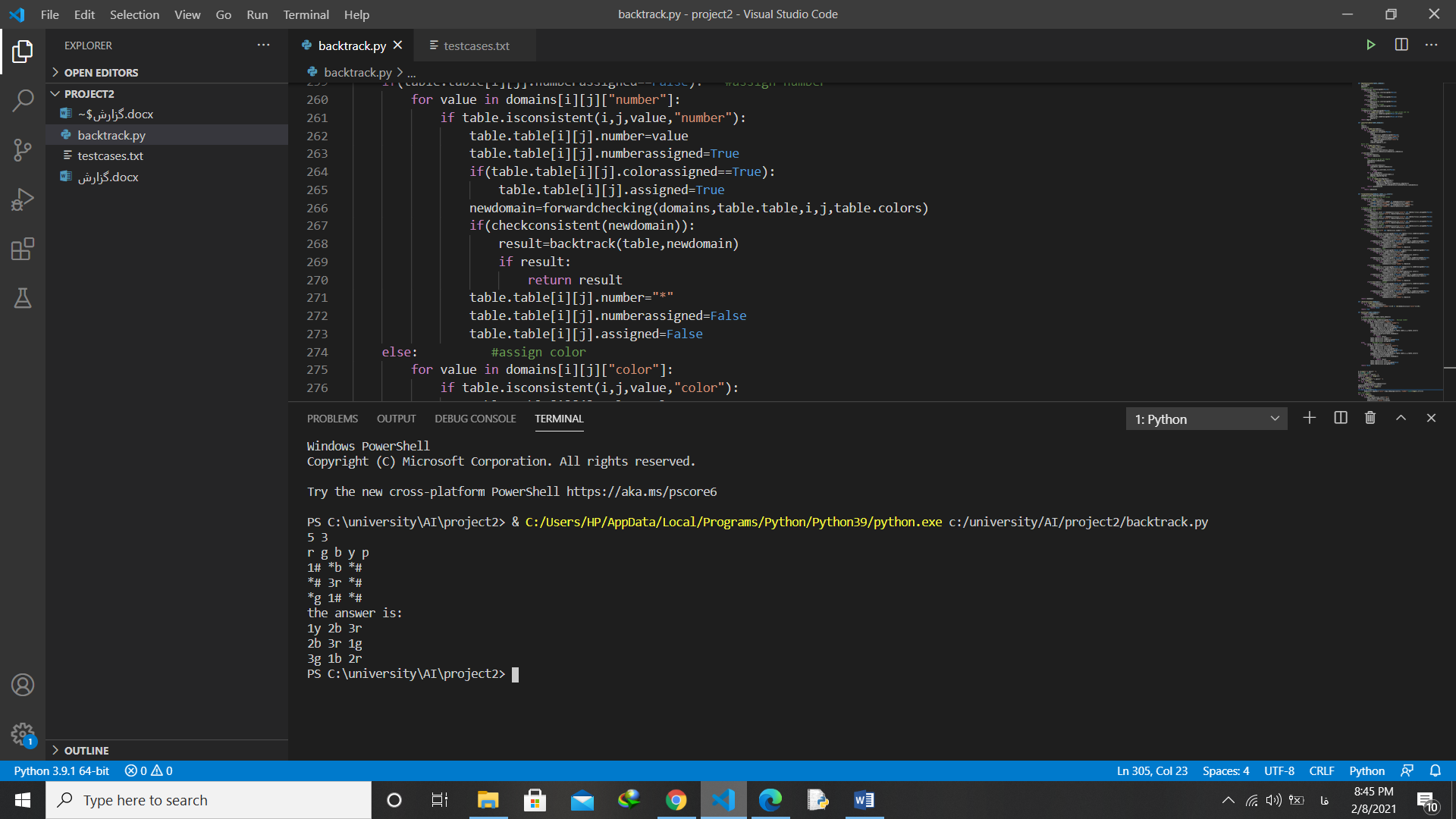
If Ni,j-1 > Ni,j => priority(Ci,j-1) > priority(Ci,j)

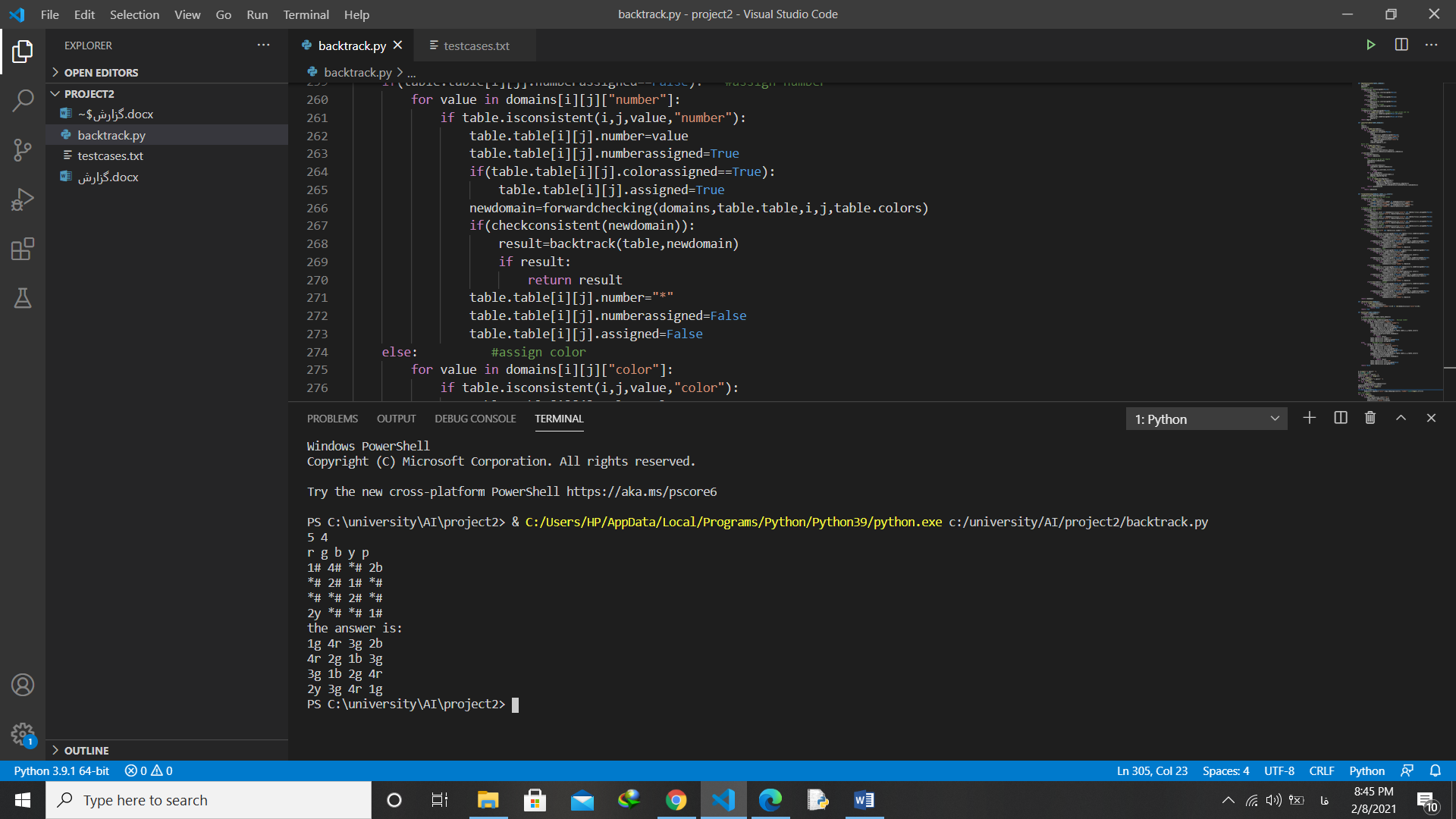
If Ni,j+1 > Ni,j => priority(Ci,j+1) > priority(Ci,j)

توضیح کد: در این کد ابتدا ورودی ها و جدول اولیه از کاربر گرفته میشود.سپس دامنه های اولیه تشکیل می شود و به تابع backtrack جدول و دامنه ها داده می شوند تا مسئله را حل کند.

در تابع backtrack که به صورت بازگشتی پیاده سازی شده است اگر جدول ورود کامل انتصاب شده باشد به عنوان پاسخ بر میگرداند در غیر اینصورت با هیوریستیک های درجه و MRV (mrv در اولویت قرار دارد) یک خانه را برای انتصاب انتخاب میکند. در تابع selectvariable درجه و طول دامنه محاسبه شده و خانه مناسب بازگردانده میشود. سپس به ازای تمام دامنه های موجود برای آن خانه مقادیر را انتصاب می دهد و با forwardchecking طبق قواعد مسئله دامنه های ناسازگار را حذف میکند و با دامنه جدید وجدول جدید مجدد backtrack استفاده می شود. اگر پاسخ داد که پاسخ بازگردانده می شود در غیر این صورت انتصاب و دامنه به حالت قبلی برمیگردد و برای مقدار جدید روند تکرار می شود.در صورتی که تمامی مقادیر دامنه استفاده شود و جواب صحیح باز نگردد تابع fail را بر میگرداند.

نمونه اجرا های test case قرار داده شده:





مسئله زیر پاسخ ندارد:

