به نام خدا

پروژه پایانی درس برنامه نویسی کامپیوتر – 4021

استاد درس: دکتر وحیده مقتدایی

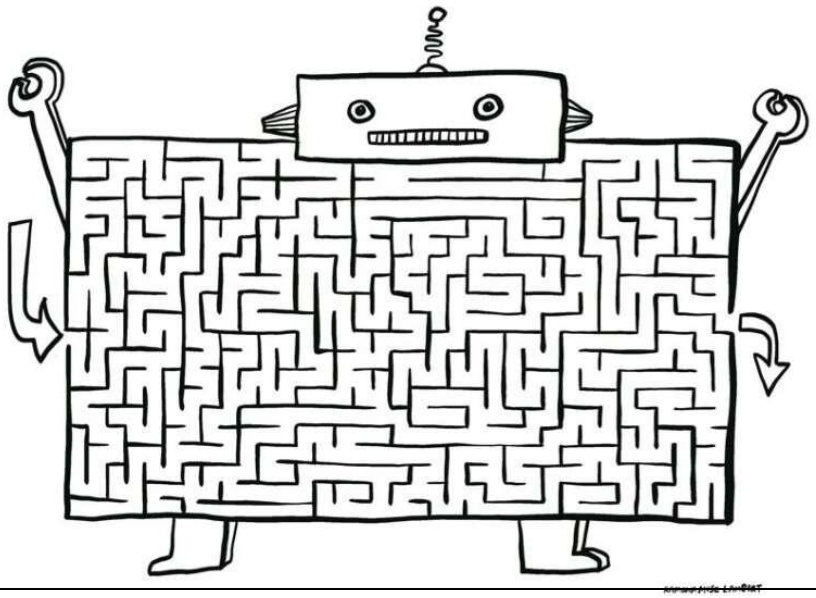
نام پروژه: مسیریاب ماز

⚠به این نکات دقت کنید:

1. قبل از انجام پروژه از نصب بودن پکیج های **random , numpy** اطمینان حاصل کنید. درصورت عدم وجود، به کمک **pip** و یا **anaconda** میتوانید آن هارا نصب کنید.
2. فایل ارسالی شما باید در یک فایل **zip.** و یا **rar.** شامل کد برنامه شما (با پسوند **ipynb.** و یا **py.**) و یک گزارش **pdf.** باشد. در گزارش خود لازم است که در خصوص نحوه عملکرد برنامه یه صورت کلی و خط به خط توضیح دهید. برای تمامی توابع نیز باید توضیح کامل داده شود.
3. نیازی به کشیدن فلوچارت نیست
4. نام فایل **zip.** یا **rar.** بایدبه صورت **"FullName_StudentNumber_MazeProject"** باشد.
5. علاوه بر ارسال فایلها، در نهایت ارایه پروژه به صورت شفاهی)در حد چند سوال و جواب(هم انجام خواهد شد.
6. کل کد شما باید در فایل **main.py** در قسمتی که با کامنت **PUT YOUR CODE FROM HERE ##** مشخص شده قرار بگیرد.
7. اصلا به کد **mazebackend.py** دست نزنید!! این کد باید حتما با **main.py** در پوشه یکسانی باشند لطفا دایرکتوری هارا تغییر ندهید تا با مشکلی مواجه نشوید.

Maze

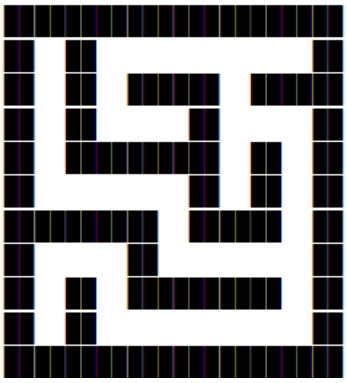
Maze ها مسیرهای تودرتویی هستند که پیدا کردن مسیر خروج را دشوار میکنند. قطعا قبلا با آن ها در بازی های فکری و یا فیلم ها مواجه شده اید. دربسیاری از آن ها چندین مسیر خروجی وجود دارد و همچنین مشخص است که اگر تفکری پشت مسیرهای انتخابی نباشد فرد قطعا گم میشود و شاید تا ابد در حلقه بماند و یک مسیر اشتباه را چندین بار برود و درنهایت مسیر خروج را نیابد. این پروژه برگرفته از این مدل مسیرهای تودرتو و یافتن الگوریتمی برای رسیدن از ورودی به خروجی با کمترین تعداد حرکت ممکن است.



- **تعریف پروژه:**

کد main.py که در فایل پروژه قرار دارد، 2 عدد فرد از کاربر ورودی میگیرد، و با توجه به این ورودی، maze ای متشکل از دیوار و مسیر باز را به صورت random، میسازد. همچنین ماتریسی را تحت عنوان "maze" خروجی میدهد. در هر دور maze ساخته شده و همچنین نقاط start و end آن در هربار اجرای برنامه، متفاوت و به صورت random ساخته میشوند. maze ساخته شده در هر بار شکلی مشابه مثال زیر دارد. بلوک های مشکی دیوار ها، و بلوک های سفید، مسیرهای باز maze هستند. در ادامه این کد به شما یک ماتریس تحت عنوان "maze" میدهد که این ماتریس متشکل از 0 برای دیوار ها، 1 برای مسیرهای باز، 2 برای start و -2 برای end هست.

```
Enter an odd number for Width: 11
Enter an odd number for Height: 11
```



```
[[ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  2.  0.  1.  1.  1.  1.  1.  1.  1.  0.]
 [ 0.  1.  0.  1.  0.  0.  0.  1.  0.  0.  0.]
 [ 0.  1.  0.  1.  1.  1.  0.  1.  1.  1.  0.]
 [ 0.  1.  0.  0.  0.  0.  0.  1.  0.  1.  0.]
 [ 0.  1.  1.  1.  1.  1.  0.  1.  0.  1.  0.]
 [ 0.  0.  0.  0.  0.  1.  0.  0.  0.  1.  0.]
 [ 0.  1.  1.  1.  0.  1.  1.  1.  1.  1.  0.]
 [ 0.  1.  0.  1.  0.  0.  0.  0.  0.  1.  0.]
 [ 0.  1.  0.  1.  1.  1.  1.  1.  1. -2.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]]
```

- **کد شما:**

- **باید در بخش مشخص شده در فایل main.py کدی بنویسید که با استفاده از ماتریس "maze" برای هر ابعادی از ماتریس که توسط کاربر وارد میشود(که الزاما مربعی نیست) مسیری از start به end خروجی دهد.**
- **در نهایت باید ماتریسی خروجی دهید مانند ماتریس "maze" و کوتاه ترین مسیر بین ورودی و خروجی را با گذاشتن عددی (متفاوت از اعداد خود ماتریس برای مثال: 3) در هر خانه مشخص کنید.**
- **در آخر باید شکلی مانند شکل بالا که در آن مسیر شما با رنگ دیگری مشخص شده است، خروجی دهید.**

راهنمایی:

برای این تسک باید ماتریس وزنی ای با ابعاد مشابه ماتریس maze تشکیل دهید و به هرخانه آن عددی نسبت دهید. عددی که به هر خانه نسبت داده می شود باید با توجه به فاصله خانه تا end باشد. (به این مدل شماره گذاری، تعیین ماتریس وزنی اقلیدسی هم گفته میشود.) در این ماتریس، هرچقدر از ورودی به سمت خروجی حرکت میکنیم باید عدد اختصاص داده شده به خانه ها، کمتر شود تا در end به صفر برسد. پس از تشکیل دادن این ماتریس باید دیوار ها را نیز در آن مشخص کنید. با برداشتن هر step در مسیر start به end باید هزینه هر step نیز درنظر گرفته شود تا بتوان به کوتاه ترین مسیر رسید. با ترکیب 3 داده بالا میتوان کمترین هزینه برای حرکت از ورودی به خروجی و درنتیجه کوتاه ترین مسیر را مشخص کرد.

!! نکته: برای اینکه بتوانید با ماتریس و maze ثابتی کد خود را تست کنید، در فایل main.py کد #random.seed(4) را از کامنت درآورده و به جای آن کد random.seed(int(random.random()*10)) را کامنت کنید. در نظر داشته باشید که این فقط برای تست هست و کد نهایی شما باید با ماتریس های random کار کند.

💥 بخش امتیازی:

کدی بنویسید که عددی از کاربر ورودی بگیرد(در range خانه های خالی) و در maze ساخته شده، از بین خانه های خالی(دیوار در آن های نباشد و نقطه شروع و پایان هم نباشند.) به تعداد آن عدد، خانه هایی را به طور random با رنگ قرمز و با عدد - 10 (هم در ماتریس و هم در شکل) مشخص کرده و مسیر دومی(علاوه بر مسیر یافته شده در مرحله قبل) را از ورودی به خروجی مشخص کند به طوری که کوتاه ترین مسیر باشد و حتما از خانه -10 عبور کند.
برای مشخص کردن این مسیر ماتریس و شکل جدیدی را خروجی دهید.