

بسم الله الرحمن الرحيم

گزارش مسئله‌ی ساخت شبکه‌ی تراوش، رنگ‌آمیزی و تشخیص وقوع تراوش

زینب ایوبی ۹۷۱۰۰۶۴۳

در این کد دو تابع اصلی وجود دارد که اولی وظیفه‌ی ساخت شبکه‌ی تراوش و رنگ‌آمیزی آن را دارد و دیگری صرفاً تشخیص می‌دهد تراوش به وقوع پیوسته یا خیر.

تابع ساخت شبکه و رنگ‌آمیزی با گرفتن طول شبکه، شبکه‌ای ۲ بعدی می‌سازد ابتدا تمام خانه‌های سمت چپ را روشن می‌کند (با جای‌گذاری عدد ۱ در آن) و تمام خانه‌های سمت راست شبکه را عدد بزرگی چون ۱۰۰۰ می‌گذارد. در ادامه با تولید اعداد تصادفی شروع به پر کردن خانه‌های شبکه با احتمال مشخصی می‌کند. (این احتمال به عنوان ورودی از کاربر گرفته می‌شود).

نحوه‌ی پر کردن خانه‌های شبکه به ترتیب از سمت چپ به راست و از بالا به سمت پایین است طوری که اگر بنا به روشن‌شدن خانه‌ای باشد برای جای‌گذاری مقدار آن به مقدار ۲ خانه‌ی سمت چپ و بالایی آن نگاه می‌شود و مطابق الگوریتم رنگ‌آمیزی کتاب اگر هر دو همسایه‌ی مجاور صفر بودند عدد جدیدی در آن قرار می‌گیرد و اگر هر دو ناصفر بودند مینیمم آن دو در خانه‌ی مورد بررسی قرار می‌گیرد و سپس همه‌ی خانه‌های هم‌مقدار با آن، مقدار جدید و مساوی با مقدار قرار داده‌شده در خانه‌ی مذکور را اتخاذ می‌کنند و در حالت نهایی اگر فقط یکی از دو همسایه غیر صفر باشد، با آن مدار غیر صفر، هم‌مقدار می‌شود.

در پایان اگر در ستون سمت راست مقدار ۱۰۰۰ موجود باشد به صفر تغییر می‌کند.

این تابع در نهایت شبکه‌ی رنگ‌آمیزی شده را خروجی می‌دهد.

تابع تشخیص وقوع تراوش، شبکه‌ی رنگ‌آمیزی شده را به عنوان ورودی می‌گیرد و در عملیات ساده‌ای بررسی می‌کند که آیا تراوش به وقوع پیوسته است یا خیر.

اگر در ستون سمت راست شبکه عدد ۱ وجود داشته باشد به معنی وقوع تراوش است و تابع عدد ۱ را خروجی می‌دهد و در غیر این صورت تراوش اتفاق نیفتاده و عدد ۰ خروجی این تابع است.

حال پس از توضیح توابع سراغ آغاز کد می‌وریم:

در ابتدا دو مقدار طول شبکه و احتمال روشن شدن خانه‌ها از کاربر به عنوان ورودی گرفته می‌شود و سپس ۲ تابع بالا به ازای این مقادیر اجرا می‌شود و در پایان وقوع یا عدم وقوع تراوش به همراه تصویر شبکه به کاربر نشان داده می‌شود:

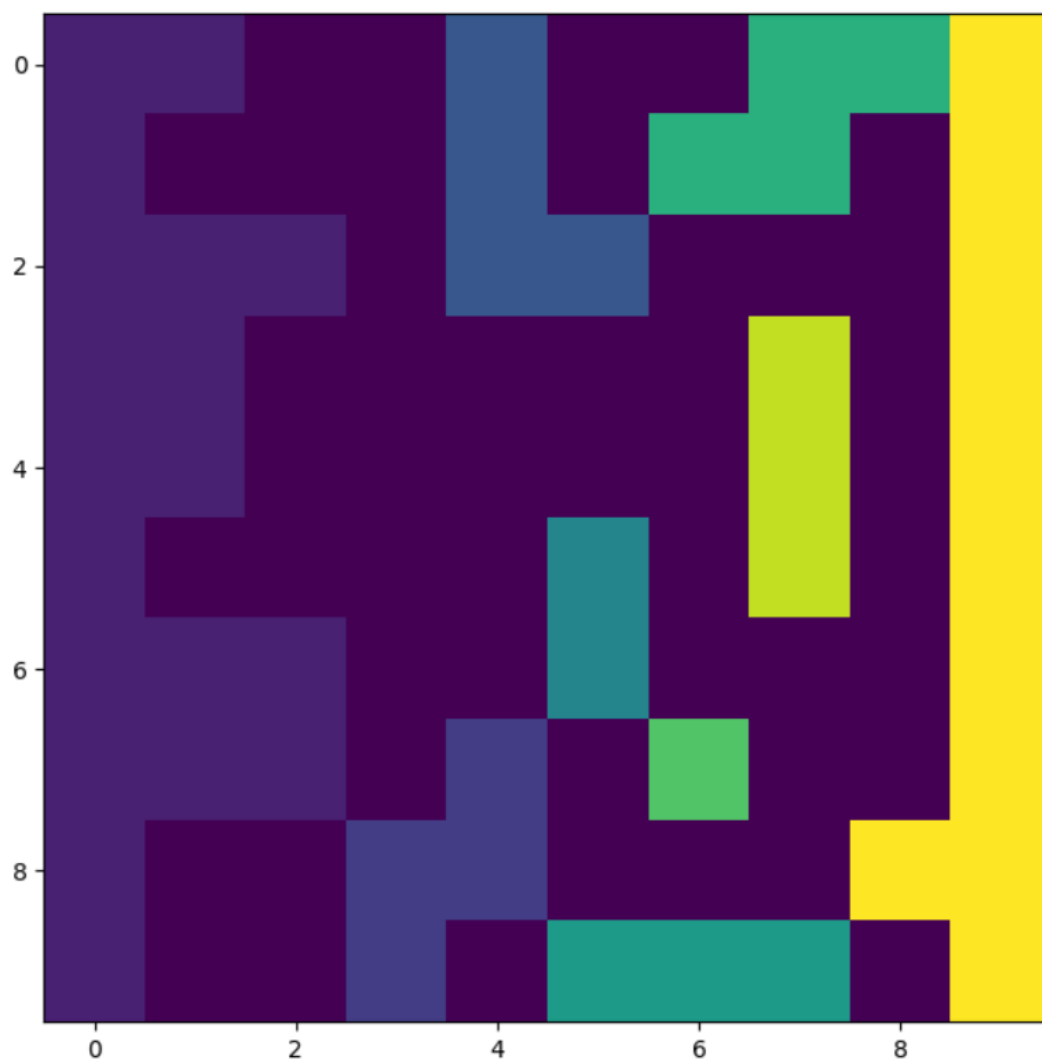
```
L = int(input("Enter the size of the lattice: "))
p = int(input("Enter the probability of turning on the lattice: ")) #p must be an integer

main_array = Creating_and_coloring_the_main_array (L , p)
R = Recognizing_the_percolation (main_array , L)

if R==0:
    print(R,"\n", "Percolation has not occured")
else:
    print(R,"\n", "Congratulations! Percolation has occured")

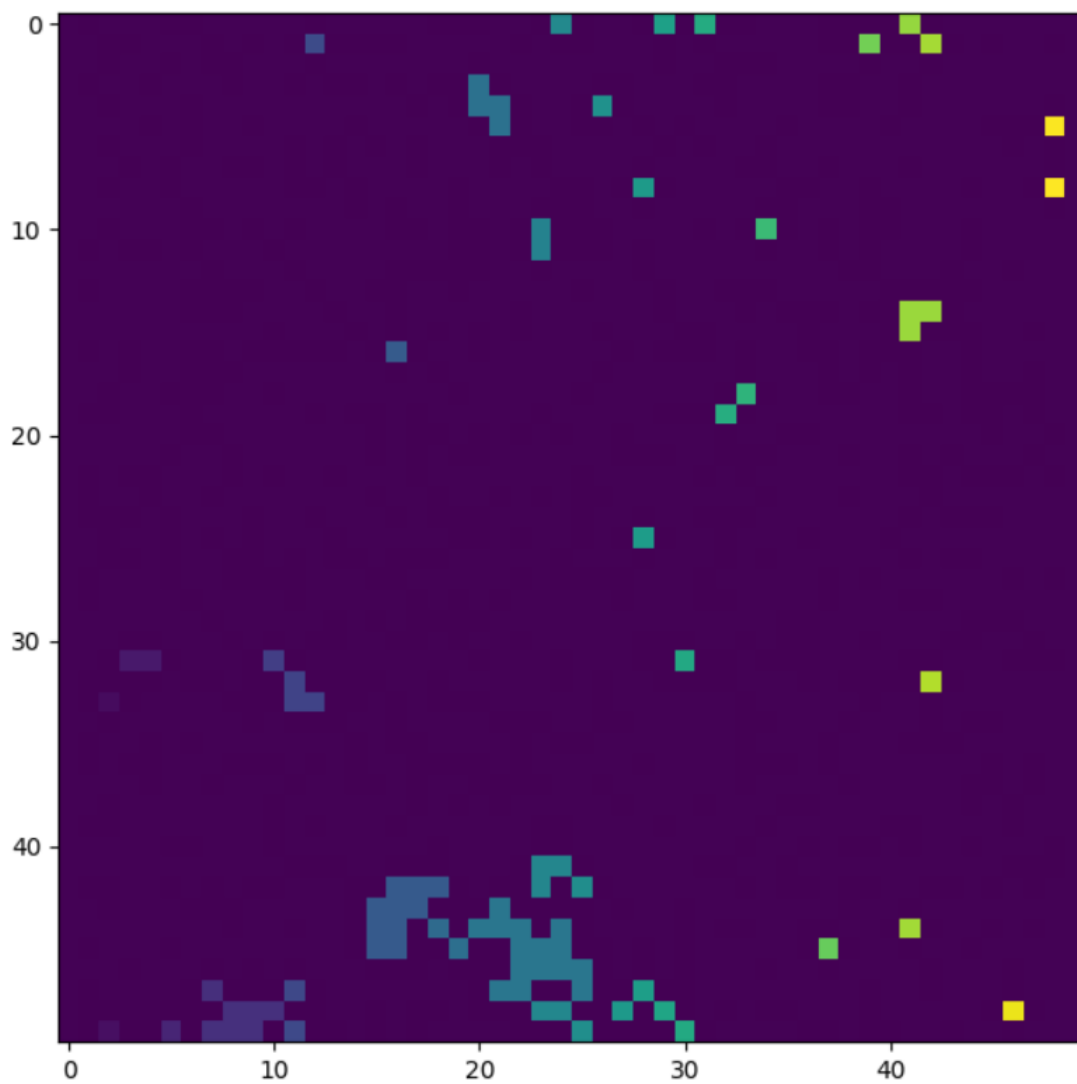
plt.imshow(main_array)
plt.show()
```

خروجی‌های گرافیکی این برنامه را برای مقادیر مختلف طول و احتمال مشاهده می‌کنید:



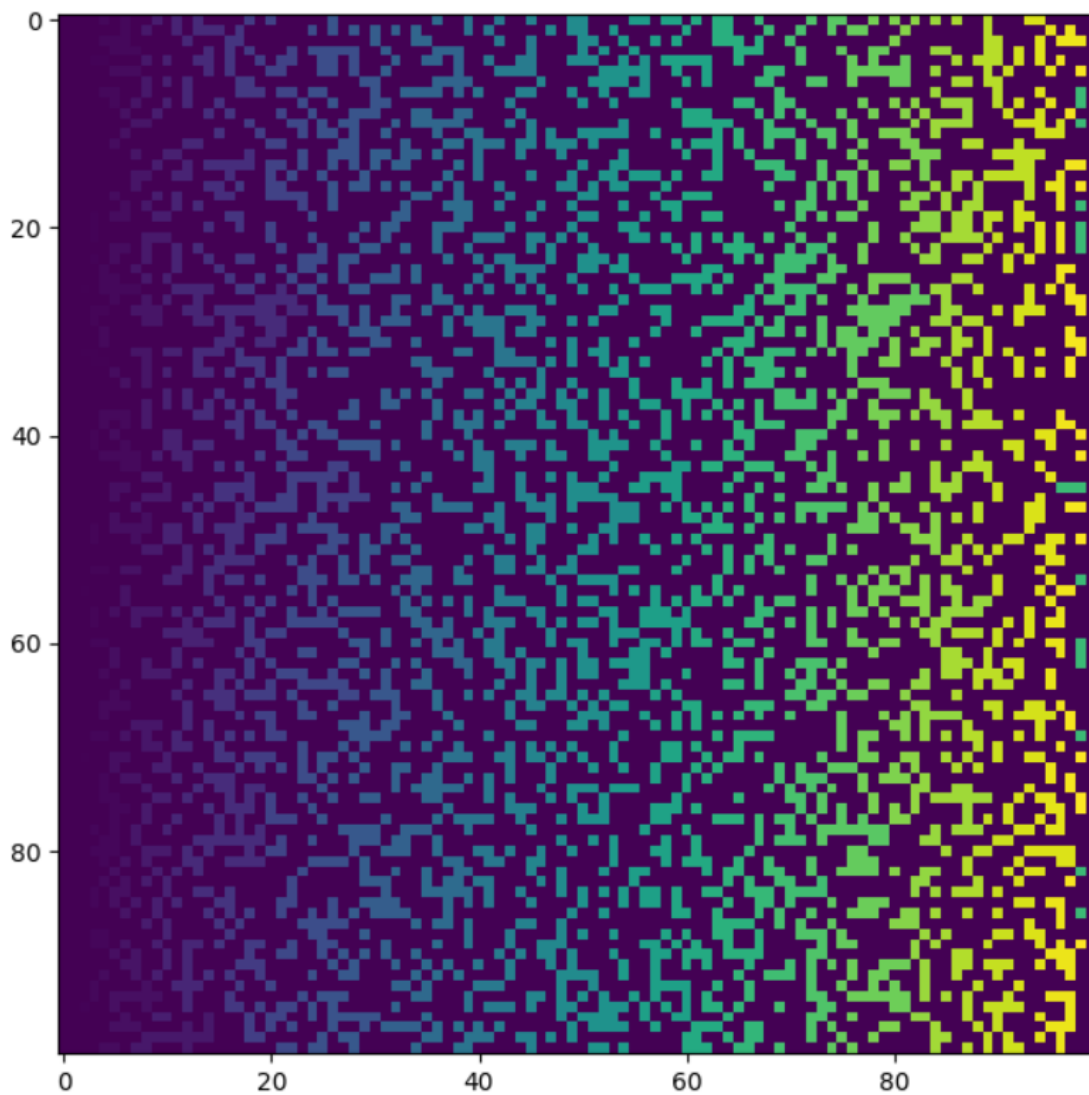
For $L=10$, $P=40$

تراوش به وقوع نپیوسته است.



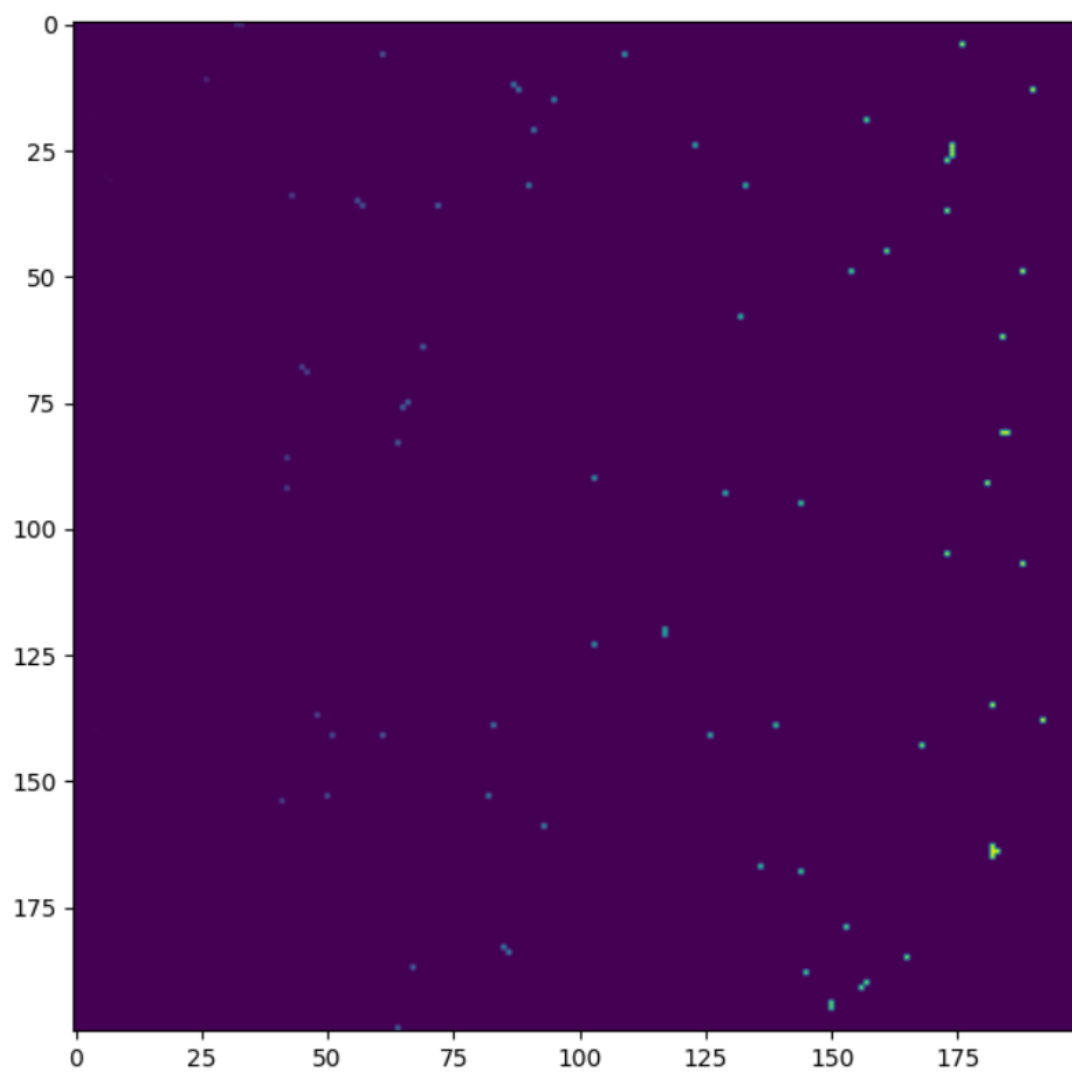
For $L=50$, $P=68$

تراوش به وقوع پیوسته است.



For $L=100$, $P=34$

تراوش به وقوع نپیوسته است.



For $L=200$, $P=78$

تراوش به وقوع پیوسته است.