

گزارش کار تمرین چهارم شبیه‌سازی رایانه‌ای در فیزیک

علی اکرامیان - 99100563

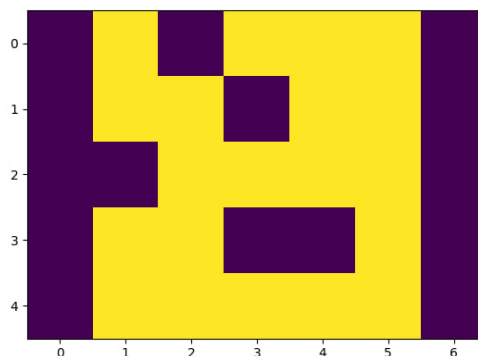
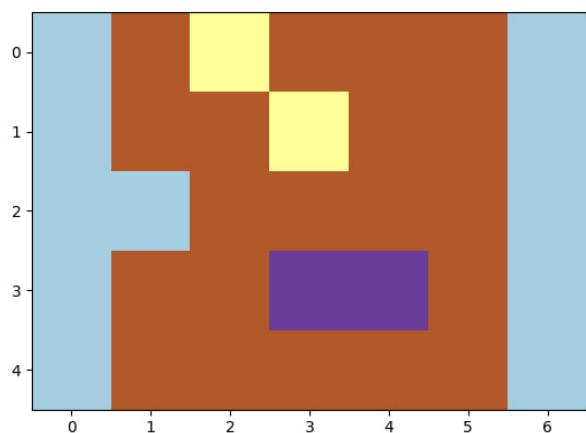
تمرین 4.2 (الگوریتم رنگ‌آمیزی):

در این تمرین من مانند الگوریتمی که سر کلاس توضیح داده شد، رنگ‌آمیزی کردم و به جواب رسیدم. کد من به این صورت است: ابتدا یک L و p که ضلع مربع و احتمال هستند را تعریف کرده‌ام و یک ماتریس mx رندوم ساخته‌ام که بین 0 و 1 است. حال یک ماتریس $color$ تعریف می‌کنم که همه‌ی درایه‌های آن صفر است. حال ستون اول و آخر را در رنگ 1 قرار می‌دهم. ستون آخر صرفاً برای زیبایی بود وگرنه کاربردی ندارد 1 کردن آن (:). حال این دو را پرینت نیز می‌کنم. حال یک حلقه‌ی دوتایی می‌زنم که تمام درایه‌های این ماتریس mx چک کند و ستون اول و آخر را دوباره 1 می‌کنم و همه‌ی دیگر درایه‌ها را چک می‌کنم که اگر کوچک‌تر از p هستند، به عدد 1 تبدیل شوند و اگر بزرگ‌تر از p هستند، 0 شوند. حال یک شمارنده‌ی $k=0$ می‌گذارم. سپس یک حلقه می‌زنم روی تمامی درایه‌های این ماتریس. ردیف اول را چک می‌کنم. اگر خودش 0 است که به ادامه‌ی حلقه بود. اگر 1 باشد چک می‌کنم که کناری آن نیز 1 هست یا نه. اگر 1 هست آن را نیز مساوی 1 می‌گذارم. اگر خودش 1 باشد ولی بغلی اش نه، به خودش یک عدد $k+1$ نسبت می‌دهم که k همان شمارنده‌ای بود که قبل از حلقه آن را 0 گذاشته بودم. حال اگر ردیف غیر از 1 بود نیز بالایی را هم چک می‌کنم که 1 هست یا نه. اگر بالایی یا چپی، n بودند، خود آن نیز n می‌شود ولی اینجا باید ببینیم اگر دو عدد مختلف در بالا و کنار آن بودند چه کنیم؟ اگر مختلف بودند، من مینیمم آن دو را حساب کردم و آن درایه را مساوی مینیمم خانه‌ی بالا و چپ می‌گذارم و بعد یک حلقه زده‌ام که همه‌ی قبلی‌های آرایه را چک کند و اگر عددی مساوی آن که مینیمم نیست پیدا کرد، آن را نیز مساوی مینیمم کند. این گونه شبکه یکی می‌شوند. و بقیه نیز مانند همان ردیف اول است. حال برای چک کردن این که تراوش رخ می‌دهد یا خیز، یک حلقه می‌زنم و ستون یکی مانده به آخر را (چون ستون آخر را خودم 1 کرده بودم) چک می‌کنم اگر 1 در آن بود که 1 پرینت کند و اگر 1 نبود نیز 0 گزارش می‌کند. یک حلقه نیز زده‌ام و 1 ها را که همان شبکه‌ی تراوش اند را مساوی یک عدد بزرگ گذاشته‌ام تا در کالرپ و رنگ بندی زیبا شود و جنبه‌ی دیگری ندارد (:)

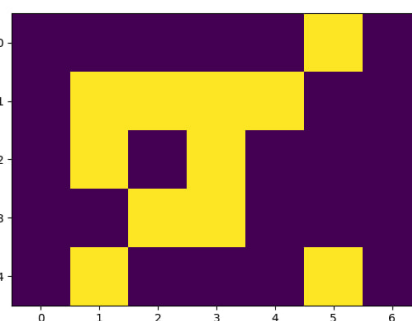
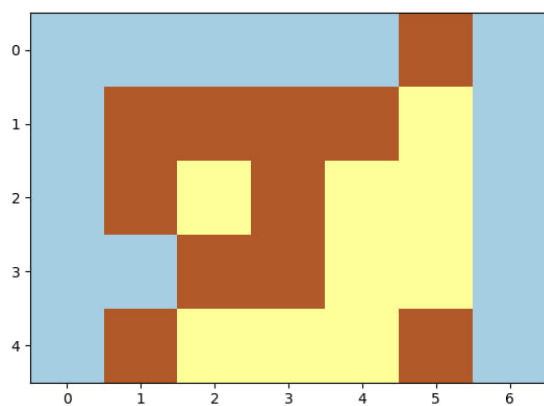
برای چندین p و L مثال می‌بینیم:

خانه‌های 1 با رنگ آبی کم رنگ و خانه‌های صفر با رنگ قهوه‌ای نشان داده شده‌اند که مثل تراوش در خاک باشد (:)
ماتریس اصلی را نیز با رنگ‌های بنفش (برای 1) و زرد (برای 0) آورده‌ام که بهتر نشان داده شود این رنگ‌آمیزی خوشه‌ها

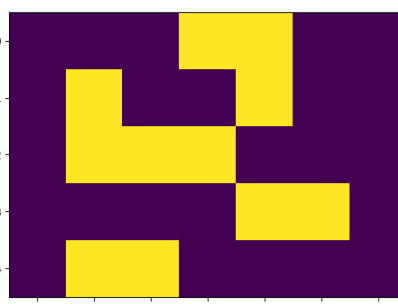
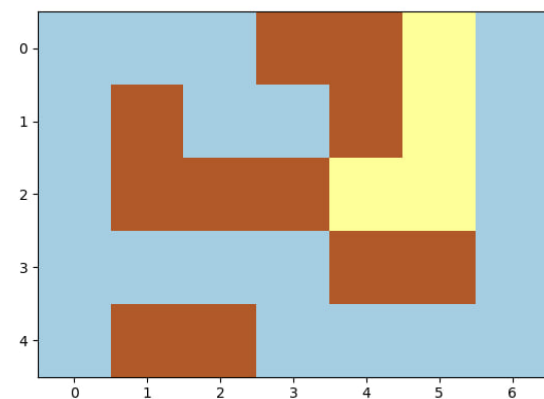
برای $L=5$ و $p=0.3$ به این صورت است:



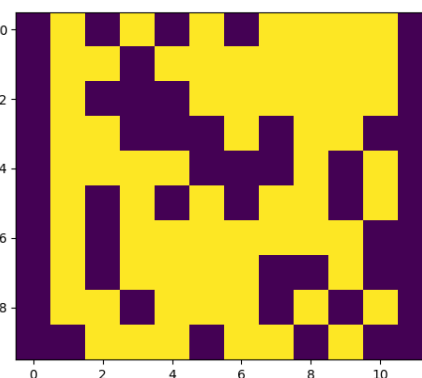
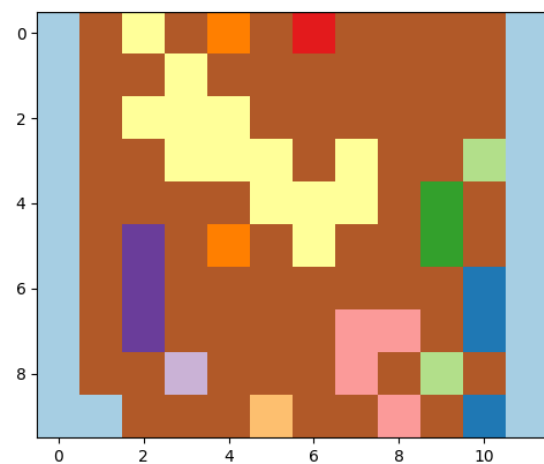
برای $L=5$ و $p=0.5$ به این صورت است:



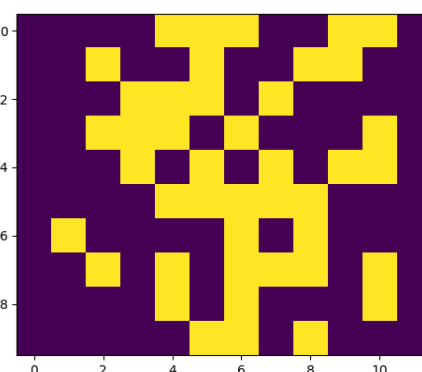
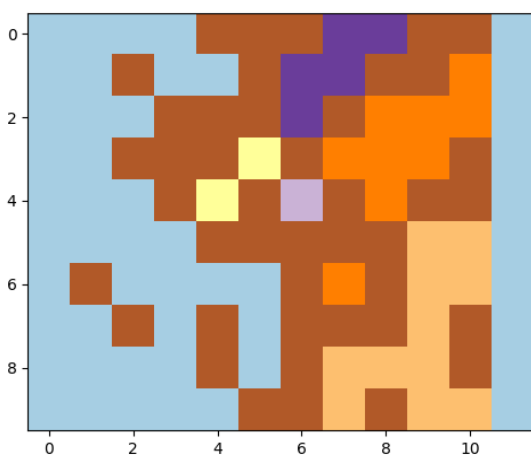
برای $L=5$ و $p=0.8$ به این صورت است:

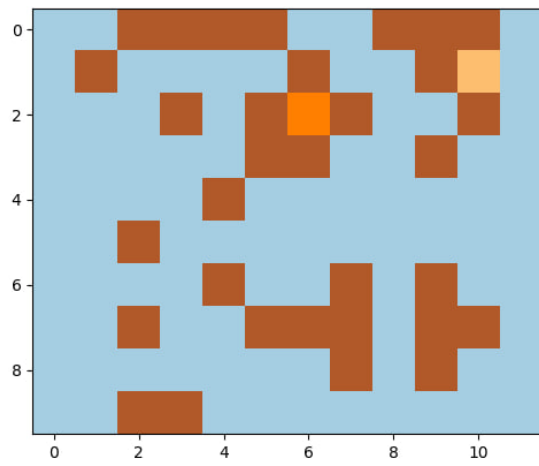


برای $L=10$ و $p=0.3$ به این صورت است:

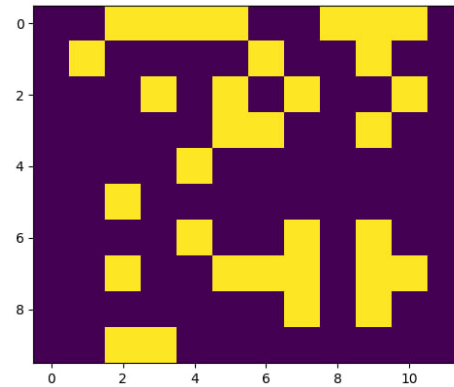


برای $L=10$ و $p=0.5$ به این صورت است:

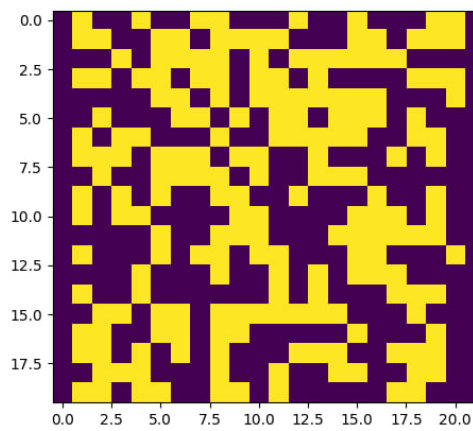
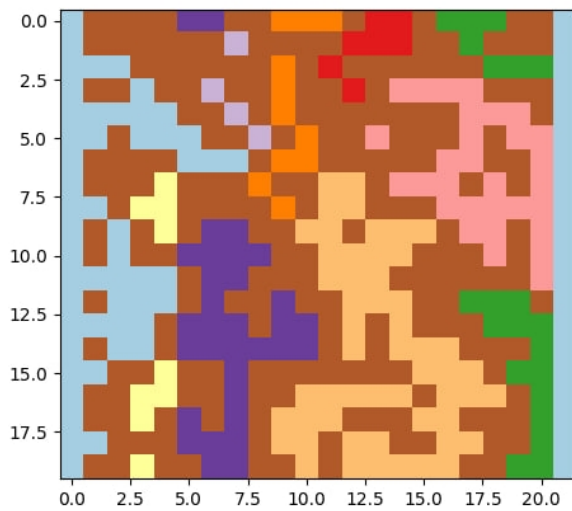




برای $L=10$ و $p=0.7$ به این صورت است:



برای $L=20$ و $p=0.5$ به این صورت است:

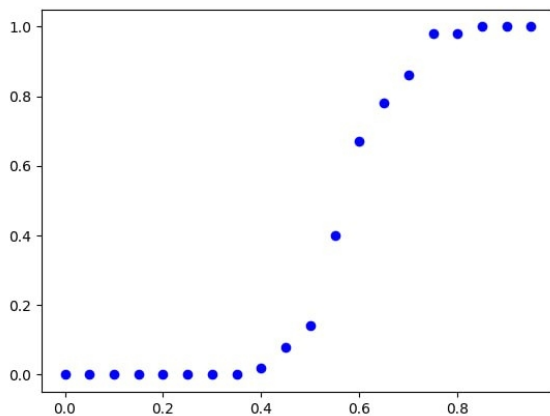


کد را نیز به دلیل طولانی بودن در اینجا نمی‌آورم و در همین فایل زیپ در فولدر تمرین 4.2 با نام `percolation.py` قرار دارد.

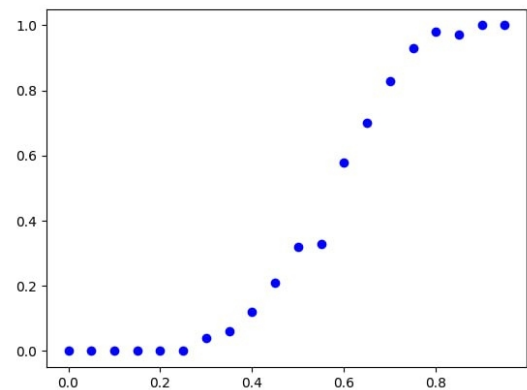
تمرین 4.3 (احتمال ایجاد خوشه‌ی بی‌نهایت برای شبکه‌ی محدود):

در این تمرین من صرفاً دو حلقه زده‌ام که یکی 20 بار اجرا می‌شود و هر بار احتمال را 0.05 بالاتر می‌برد (از 0 تا 1) و یکی دیگر که درون این حلقه قرار دارد، 100 بار این را اجرا می‌کند و میانگین می‌گیرد. برای ذخیره کردن داده‌ها و سپس رسم آنها نیز من یک آرایه‌ی `pr` دارم که احتمال‌ها را داخل آن میریزم هر بار که حلقه‌ی بیرونی تر اجرا می‌شود و یک آرایه‌ی `perc_arr` هم دارم که اول هر بار آن را صفر کرده و سپس حلقه‌ی درونی اجرا می‌شود و 0 و 1‌ها (رخداد تراوش) درون آن ریخته شده و سپس داده‌ها ازین آرایه گرفته شده و میان‌گیری می‌شود و سپس داخل آرایه‌ی `perc_pr` ریخته می‌شود که میانگین تراوش (بین 0 و 1) را برای هر احتمال را دارد. حال `perc_pr` را بر حسب `pr` رسم می‌کنیم و داریم:

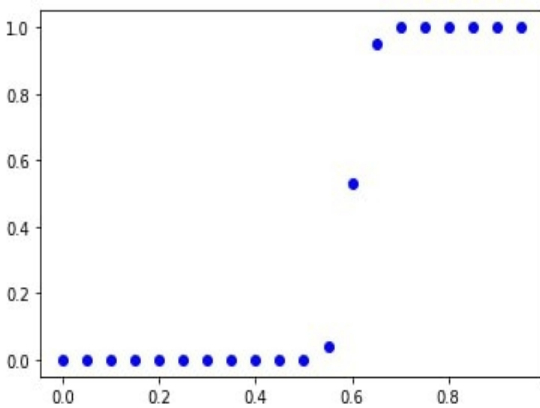
حال برای چند طول می‌بینیم:



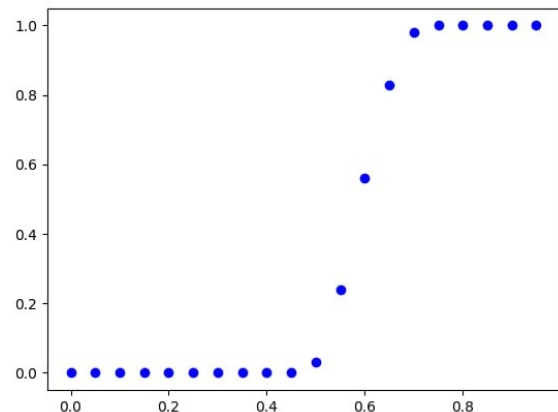
L = 10



L = 5

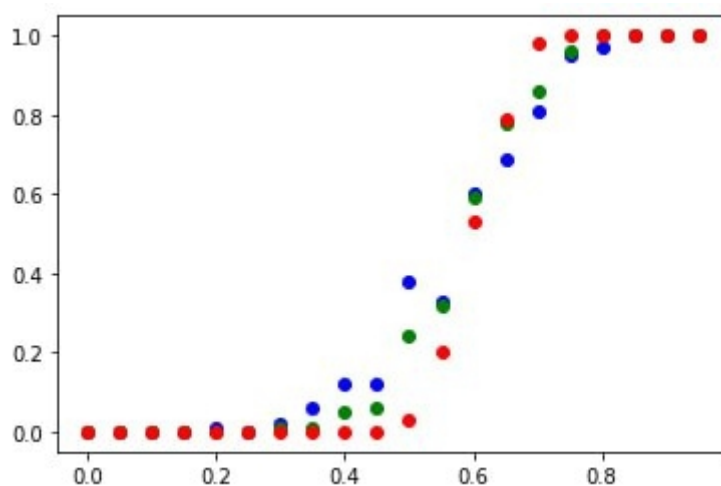


L = 50

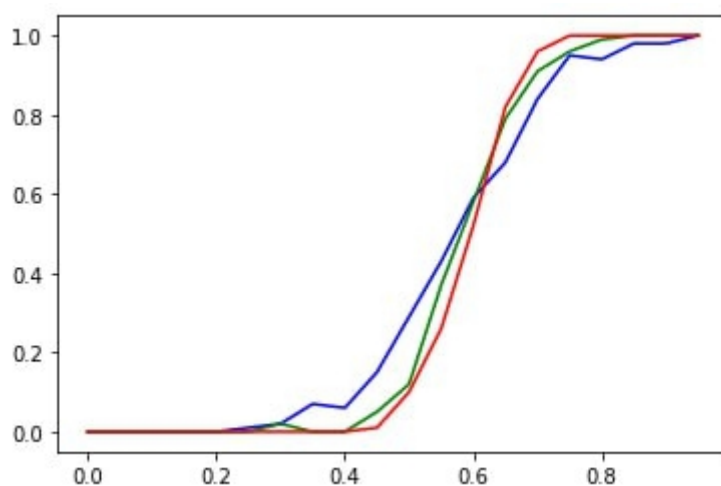


L = 20

حال در یک جا کنار هم می بینیم:



یا اگر خط رسم کنیم خواهیم داشت:

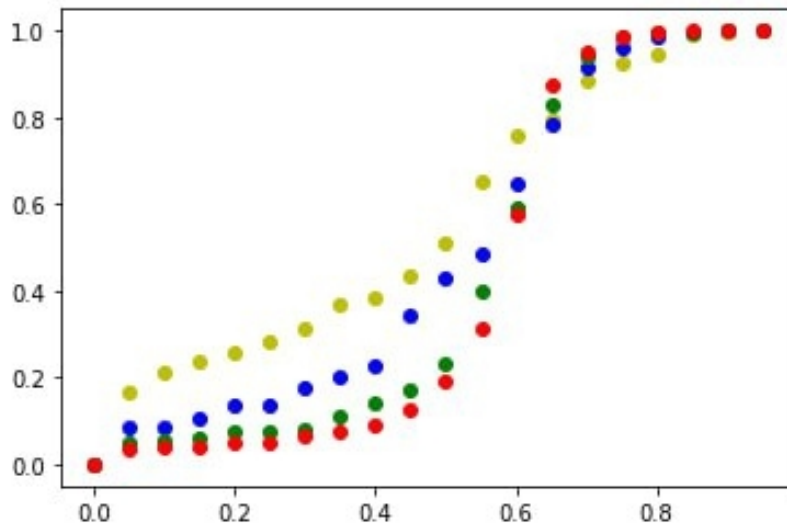


کد را نیز به دلیل طولانی بودن در اینجا نمی آورم و در همین فایل زیپ در فولدر تمرین 4.2 با نام percolation-pr.py قرار دارد.

تمرین 4.4 (احتمال اتصال به خوشه‌ی بی‌نهایت):

در این تمرین نیز از همان کد قبلی استفاده کرده و صرفاً یک آرایه Q تعریف می‌کنیم و هر بار کمیت تعداد 1 ها در آرایه را تقسیم بر تمامی درایه‌های غیر آن (کل خانه‌هایی که 1 شده است ممکن است در شبکه‌ی بی‌نهایت باشند یا نباشند) می‌کنیم و این آرایه را در نهایت بر حسب همان pr می‌کشیم.

حال نتایج را برای 4 طول $L = 5, 10, 20, 30$ در یک شکل می‌بینیم. خواهیم دید که هرچه طول بیشتر می‌شود شیب نمودار در وسط آن شدیدتر شده و یکپهویی تر به 1 سعی می‌کنند برسند. (شیب خیلی زیادتر می‌شود در طول‌های خیلی بزرگ)



کد را نیز به دلیل طولانی بودن در اینجا نمی‌آورم و در همین فایل زیپ در فولدر تمرین 4.4 با نام `percolation-Qinfity.py` قرار دارد.

من در هر کدام از کدها در گوگل کولب هم زده‌ام تا سریع‌تر ران شوند و نمودارهای چندتایی همه در فایل‌ها با همان اسم فقط به صورت `ipynb` هستند. در فایل‌های `py` که هستند، نمودارها تکی رسم می‌شوند.