

بسم الله الرحمن الرحيم

گزارش طول همبستگی (شعاع ژیراسیون) تراوش

زینب ایوبی ۹۷۱۰۰۶۴۳

این بار برای ساختن آرایه‌ی اصلی تراوش از الگوریتم هشن-کپلمن استفاده کرده‌ام سپس برای طول‌های مختلف یک‌بار متوسط اندازه‌ی خوشه‌های غیر بی‌نهایت و بار دیگر شعاع ژیراسیون یعنی طول همبستگی سیستم را برای خوشه‌های غیر بی‌نهایت محاسبه نمودم. (بدین منظور ۲ فایل کد ارائه شده‌است).

ابتدا به تشریح و نتایج کد اول با نام `Length of correlation (4.5 , 4.6)` می‌پردازم:

این کد از چندین تابع تشکیل شده‌است.

اولین تابع با نام `Hoshen_Keopلمان(L, p)` با گرفتن طول سیستم و احتمال روشن کردن هر خانه با استفاده از الگوریتم هشن-کپلمن که در کتاب توضیح داده شده‌است (و من برای جلوگیری از تکرار مکررات مجدد آن را توضیح نمی‌دهم) آرایه‌ی اصلی تراوش را ایجاد می‌کند. خروجی این تابع ۳ چیز است: اول آرایه‌ی اصلی تراوش، دوم آرایه‌ای با نام `S` که اندازه‌ی تمامی خوشه‌ها را نگه می‌دارد مثلاً `S[1]` تعداد خانه‌هایی از آرایه‌ی اصلی را نشان می‌دهد که `index` آن‌ها ۱ است یعنی عضو خوشه‌ی ۱ هستند و سومین خروجی آن آرایه‌ی `map_array` است که نشان می‌دهد هر عدد داخل آرایه‌ی اصلی به چه عددی اشاره می‌کند. (در واقع خروجی سوم تابع زائد است زیرا در داخل خود تابع با استفاده از دو حلقه‌ی `for` در قسمت انتهایی تابع مقادیر آرایه‌ی اصلی با مقدار واقعی خود جای‌گزین می‌شوند و پس از آن هر عدد داخل آرایه‌ی اصلی به خود اشاره می‌کند. نکته‌ی قابل ذکر این که برای خواندن آرایه‌ی `map_array` و جای‌گذاری اعداد داخل آرایه با مقدار واقعی خود از تابعی بازگشتی با نام `proc(k, w)` استفاده می‌شود که بدین صورت کار می‌کند: اگر شماره‌ی ستون آرایه‌ی `map_array` با `index` آن برابر بود یعنی عدد به خودش اشاره می‌کند خروجی تابع همان عدد خواهد بود و در غیر این صورت تابع با صدا زدن خودش فرآیند را این بار برای `index` ستونی تکرار می‌کند که عدد قبلی به آن اشاره می‌کرد).

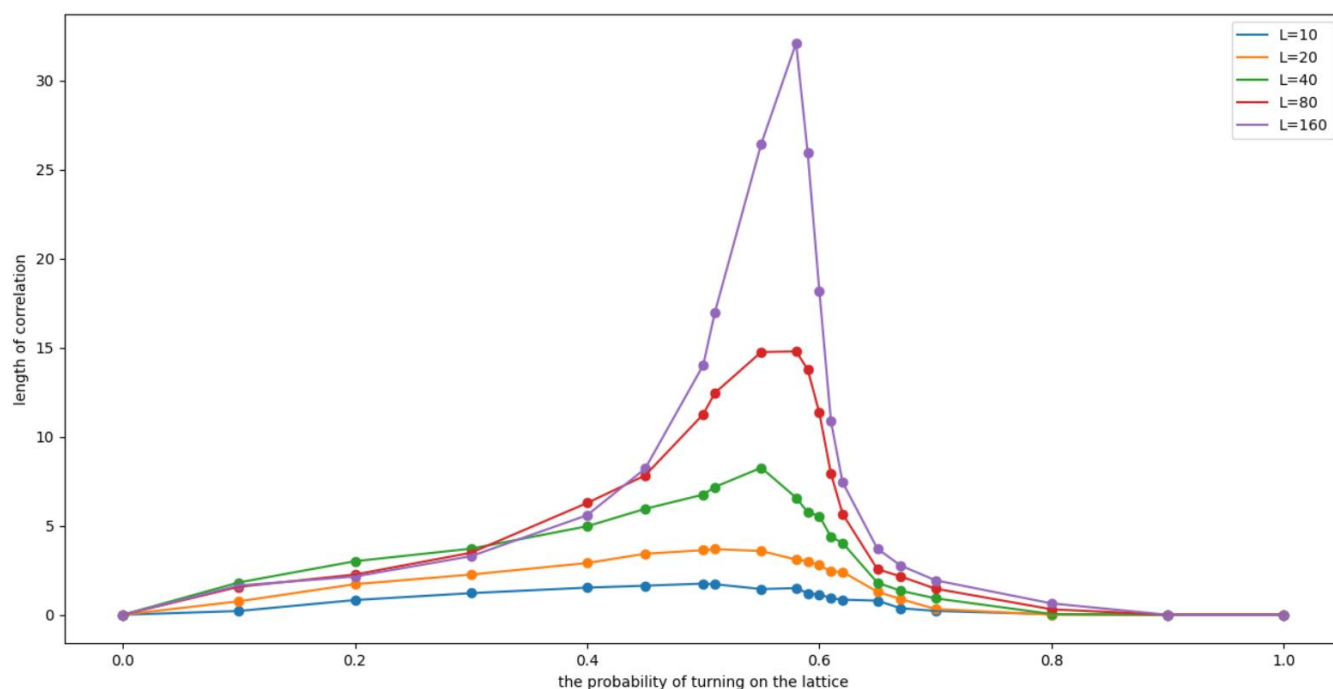
تابع بعدی تابع `Recognizing_the_percolation(A, L, map_array)` است که وقوع تراوش را به سادگی تشخیص می‌دهد. با گرفتن آرایه‌ی ساخته‌شده از خروجی تابع قبلی بررسی می‌کند که آیا در ستون آخر آن ۱ یا عددی که به ۱ اشاره می‌کند موجود است یا خیر. در صورت وجود تراوش به وقوع پیوسته و در غیر این صورت خیر.

تابع بعدی `length_of_correlation(L, p)` است که برای یک طول سیستم مشخص و احتمال روشن کردن خانه‌های شبکه‌ی p شعاع ژیراسیون دومین خوشه‌ی بزرگ غیر بی‌نهایت را حساب کرده و خروجی می‌دهد. برای یافتن دومین خوشه‌ی بزرگ غیر بی‌نهایت از آرایه‌ی S که اندازه‌ی خوشه‌ها را نگهداری می‌کند بهره می‌برد.

نکته‌ای در انتخاب دومین خوشه‌ی بزرگ غیر بی‌نهایت بجای اولین آن نهفته است و آن این است که چون در الگوریتم هشن کیلمن ستون سمت چپی کنار آرایه‌ی اصلی تماماً ۱ می‌شود در اولین ستون سمت چپ آرایه عددی جز ۱ نمی‌تواند قرار بگیرد و به همین علت بعضی خوشه‌ها به غیر ضرورت به خوشه‌ی ۱ می‌پیوندند و این باعث بزرگ‌شدن نامطلوب خوشه‌ی شماره‌ی ۱ می‌شود. برای رفع این مشکل با کمک آقای معمر دومین خوشه‌ی غیر بی‌نهایت مورد بررسی قرار گرفت که نتایج مطلوبی را تولید خواهد کرد که به زودی نظاره می‌کنید.

و تابع دیگر تابع `running(L, p)` است که برای هر L و p مشخص ۱۰۰ بار طول همبستگی سیستم را بدست با صدا زدن تابع `length_of_correlation(L, p)` بدست آورده و متوسط آن را خروجی می‌دهد.

قسمت انتهایی کد برای رسم نمودار خروجی‌های توابع است که نیازی به توضیح ندارد و اکنون می‌توانیم خروجی‌های گرافیکی برنامه را مشاهده کنیم:



نمودار طول همبستگی (شعاع ژیراسیون) شبکه به ازای طول‌های مختلف بر حسب احتمال روشن کردن خانه‌های شبکه

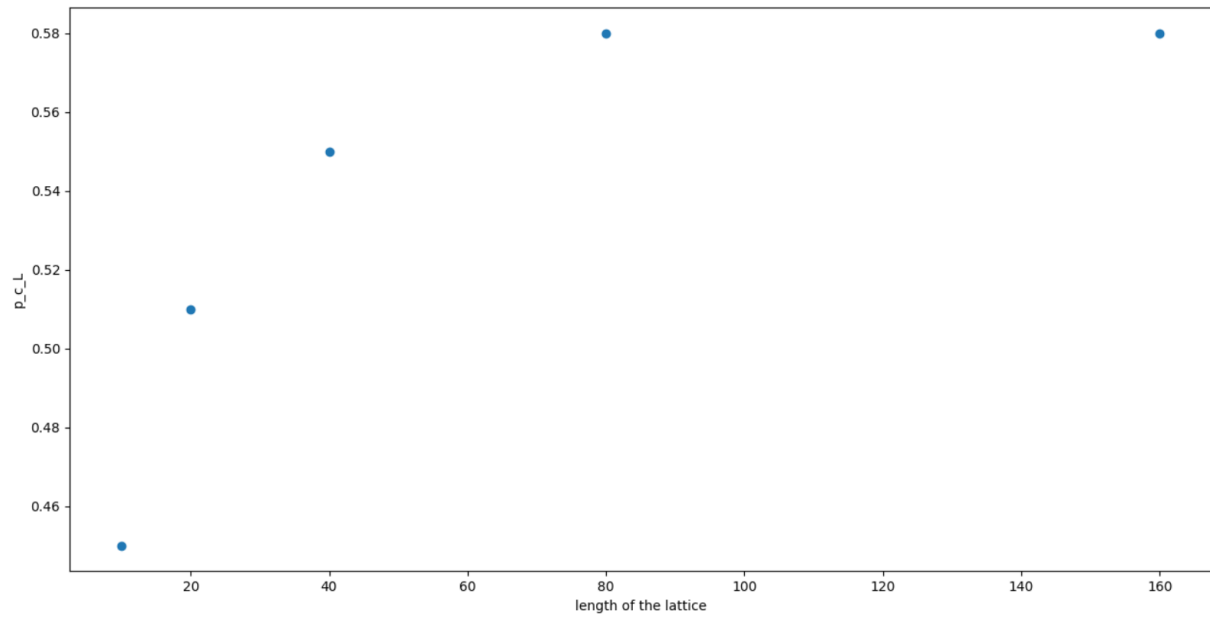
همان‌طور که انتظار داریم هر چه اندازه‌ی شبکه افزایش می‌یابد تراوش دیرتر به وقوع می‌پیوندد و طول همبستگی شبکه افزایش می‌یابد. هم‌چنین پس از وقوع تراوش و با افزایش احتمال روشن شدن خانه‌های شبکه خوشه‌ها یکی یکی به خوشه‌ی بی‌نهایت می‌پیوندند و طول همبستگی سیستم به سمت صفر می‌رود. $P_c(L)$ را برای طول‌های مختلف مشاهده می‌کنید:

$$L = [10, 20, 40, 80, 160]$$

$$P_c(L) = [0.45, 0.51, 0.55, 0.58, 0.58]$$

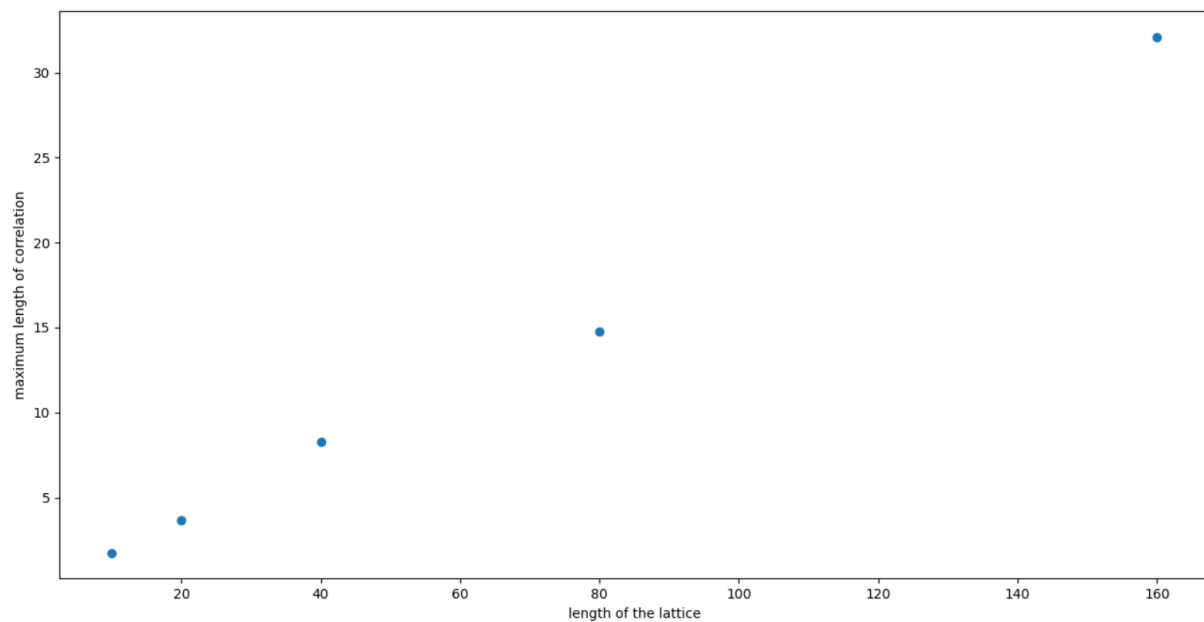
$P_c(L)$ احتمالی است که اگر خانه‌های شبکه‌ای با اندازه‌ی L با آن روشن شوند تراوش به وقوع می‌پیوندد.

برای مشاهده‌ی بهتر نمودار دو آرایه‌ی بالا را بر حسب هم رسم کرده‌ام:



PC(L) بر حسب طول شبکه

همچنین نمودار ماکسیمم طول هم‌بستگی هر سیستم بر حسب طول شبکه به صورت زیر است:



نمودار ماکسیمم طول هم‌بستگی بر حسب طول شبکه

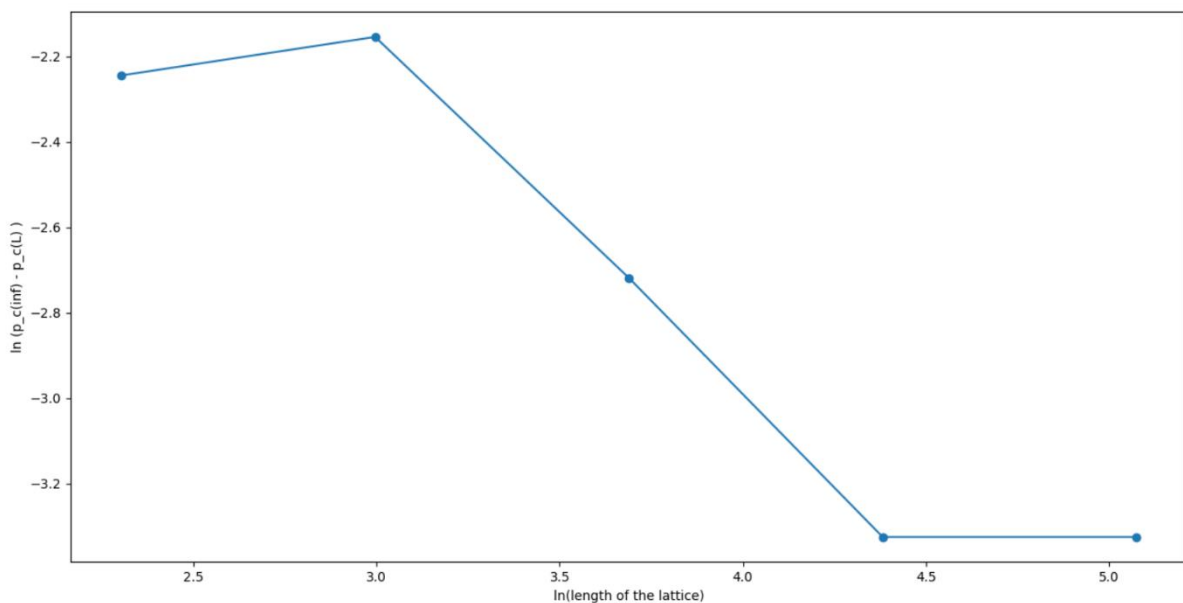
حال به قسمت انتهایی کد می پردازیم:

جایی که با فیت کردن یک منحنی بر نمودار رشد $P_c(L)$ بر حسب طول شبکه، مقدار $P_c(\text{infinite})$ را حدود ۰,۶ بدست آوردم.

$$P_c(\infty) = 0.6014619387184545 \pm 0.024179171898472906$$

سپس با رسم نمودار $\ln(|P_c(\infty) - P_c(L)|)$ بر حسب $\ln(L)$ و با استفاده از شیب نمودار (منفی معکوس شیب نمودار) کمیت ν را حدود ۲ بدست آوردم:

$$\nu = 2.0808915344804632$$



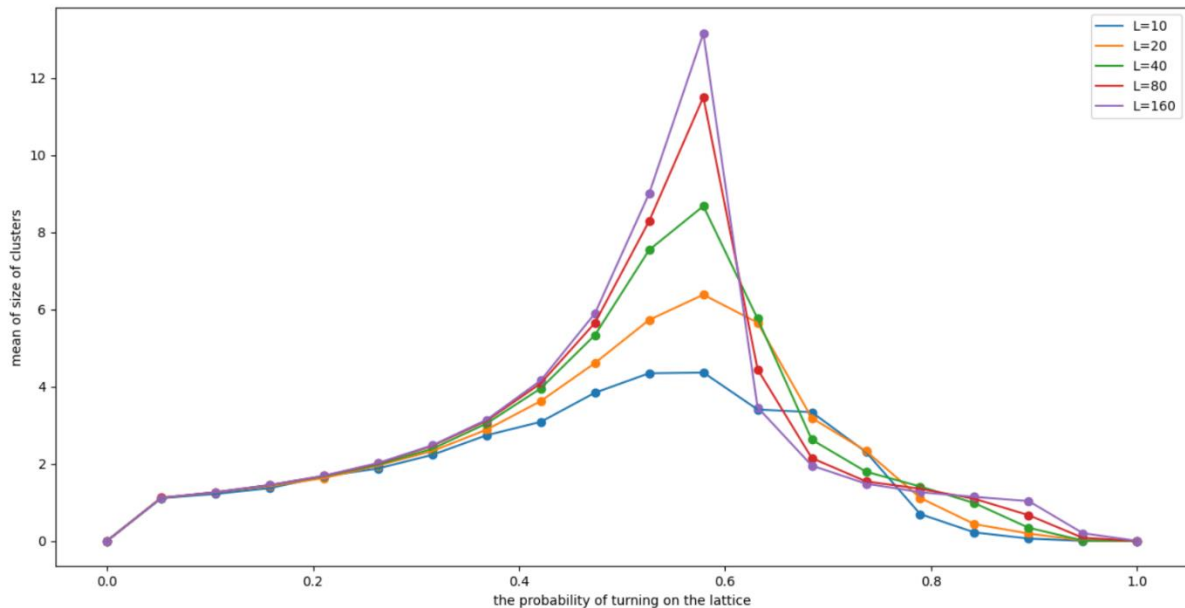
نمودار لگاریتم طبیعی قدر مطلق اختلاف $P_c(L)$ و $P_c(\text{inf})$ بر حسب لگاریتم طبیعی طول شبکه

۲ نکته‌ی اضافی:

۱. من کد بالا را بیش از ۱۰، ۲۰ بار ران کردم و گاهی نتایج بدی گرفتم مثلاً یکی دوبار $P_c(\text{infinite})$ را ۱,۴ بدست آوردم! و ندانستم چرا! (:

البته اکثر اوقات بین ۰,۶ تا ۰,۶۵ بدست آوردم.

۲. مطلب بعد بدست آوردن میانگین اندازه‌ی خوشه‌های غیربی‌نهایت به عنوان معیاری از طول مشخصه‌ی سیستم است که ابتدای گزارشم وعده‌ی آن را داده بودم:



نمودار متوسط اندازه‌ی خوشه‌های غیر بی‌نهایت بر حسب احتمال روشن کردن خانه‌های شبکه برای طول‌های مختلف

همان‌طور که انتظار داریم هر چه اندازه‌ی شبکه افزایش می‌یابد تراوش دیرتر به وقوع می‌پیوندد و متوسط اندازه‌ی خوشه‌های غیر بی‌نهایت شبکه افزایش می‌یابد. هم‌چنین پس از وقوع تراوش و با افزایش احتمال روشن شدن خانه‌های شبکه خوشه‌ها یکی یکی به خوشه‌ی بی‌نهایت می‌پیوندند و متوسط اندازه‌ی خوشه‌های غیر بی‌نهایت سیستم به سمت صفر می‌رود.

این نمودار خروجی کد دوم با نام `mean of the size of the clusters` است که دقیقاً

ساختاری شبیه به کد اول دارد. تنها تفاوت آن با کد پیشین جای‌گزینی تابع

`calculating_the_mean_of_size_of_clusters(L, p)` به جای تابع

`length_of_correlation (L , p)` است. این تابع با استفاده از آرایه‌ی S (یادآوری می‌کنم S

آرایه‌ای بود که اندازه‌ی خوشه‌ها را در خود نگه‌می‌دارد.) میانگین اندازه‌ی خوشه‌هایی با شماره‌ی ۲ و بالاتر را بدست آورده خروجی می‌دهد.

یادآوری: نکته‌ای در شروع از دومین خوشه‌ی بزرگ غیر بی‌نهایت بجای اولین آن نهفته است و آن این است که چون در الگوریتم هشن کیلمن ستون سمت چپی کنار آرایه‌ی اصلی تماماً ۱ می‌شود در اولین ستون سمت چپ آرایه عددی جز ۱ نمی‌تواند قرار بگیرد و به همین علت بعضی خوشه‌ها به غیر ضرورت به خوشه‌ی ۱ می‌پیوندند و این باعث بزرگ‌شدن نامطلوب خوشه‌ی شماره‌ی ۱ می‌شود.

رشد خوشه

کد این سوال کد 4.7 است.

این کد از چند تابع تشکیل شده‌است:

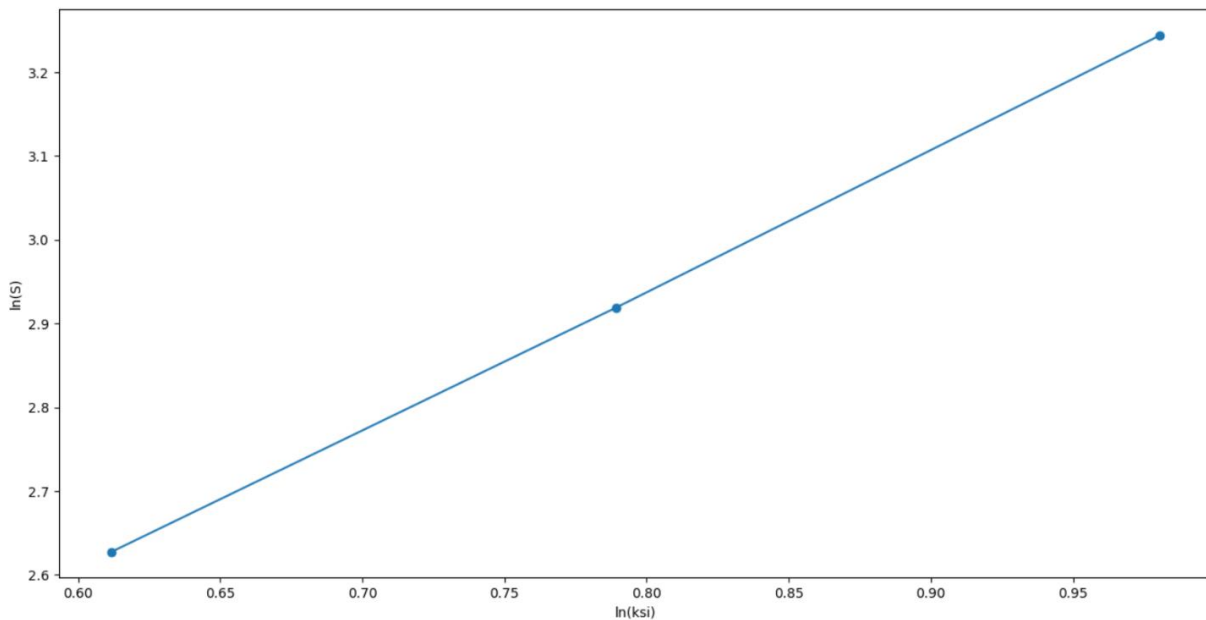
تابع اول با نام `creating_the_cluster (A , L , p)` تابعی بازگشتی است که آرایه‌ی خالی A با اندازه‌ی $L * L$ که تنها نقطه‌ی مرکزی آن روشن است را ورودی می‌گیرد و از آن‌جا شروع کرده برای هر نقطه‌ی روشن با احتمال p ۴ همسایه‌اش را روشن می‌کند. همسایگانی که روشن می‌شوند مقدار ۱ و آن‌هایی که مسدود می‌شوند مقدار ۲ می‌گیرند. آن خانه‌ای که همسایگانش بررسی شده‌اند مقدار ۱۱ می‌گیرد تا بار دیگر تابع به آن بازنگردد. در نهایت تابع به طور بازگشتی تمام همسایگان ممکن را روشن می‌کند تا جایی که دیگر همسایه‌ای امکان روشن شدن نداشته باشد یعنی تمام همسایگان موجود مسدود شده باشند. در پایان آرایه‌ی A را که خوشه در آن رشد کرده است خروجی می‌دهد. (در این تابع من شرایط مرزی را پریودیک در نظر گرفته‌ام یعنی انگار بالا و پایین و چپ و راست شبکه به هم متصل هستند).

تابع بعدی با نام `calculating_S (A)` آرایه‌ی تکمیل‌شده‌ی A را می‌گیرد و تعداد خانه‌های روشن آن را شمرده و در متغیر S به عنوان اندازه‌ی خوشه ذخیره می‌کند و S را خروجی می‌دهد.

تابع بعدی `length_of_correlation (A , L , S)` طول همبستگی (شعاع ژیراسیون) را برای خوشه‌ی رشدیافته با یافتن مرکز جرم خوشه و سپس محاسبه‌ی جذر متوسط مجذور فواصل عناصر خوشه از مرکز جرم آن، بدست آورده و خروجی می‌دهد.

و در پایان تابع $\text{running}(L, p)$ با ۱۰۰ بار اجرای کد و متوسط‌گیری، مقادیر متوسط S و طول همبستگی شبکه (ksi) را خروجی می‌دهد.

در پایان کد را برای ۳ احتمال ۰,۵ و ۰,۵۵ و ۰,۵۹ اجرا و نمودار $\ln(S)$ را بر حسب $\ln(\text{ksi})$ رسم کرده که بسیار نزدیک به یک خط است و من شیب این خط را بین ۱,۵ تا ۱,۸ بدست می‌آورم.



نمودار لگاریتم طبیعی اندازه‌ی خوشه‌ی رشد یافته بر حسب لگاریتم طبیعی طول همبستگی شبکه