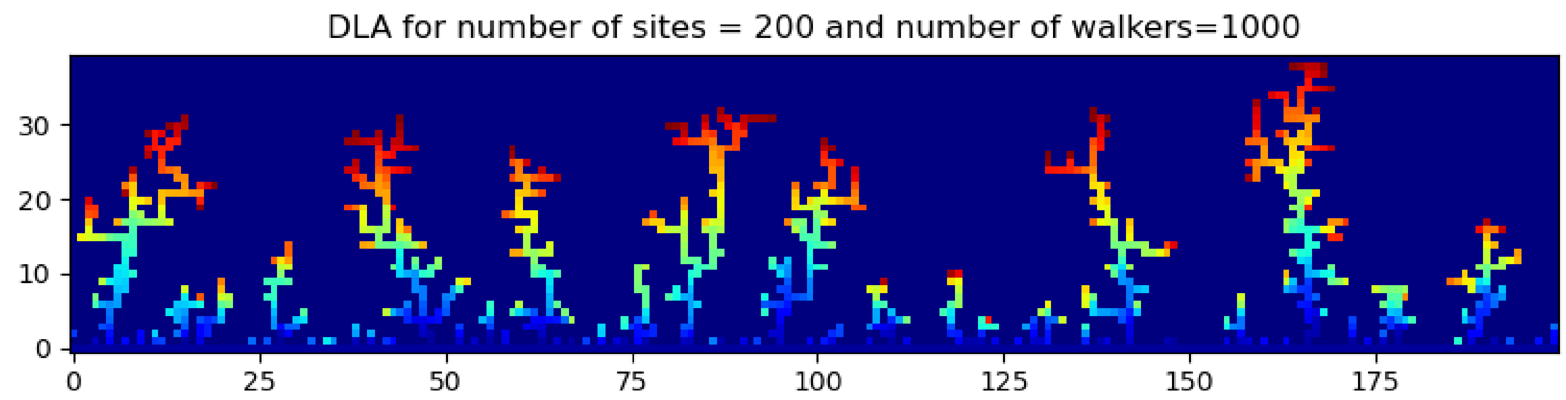
*گزارش تمرین پنجم*

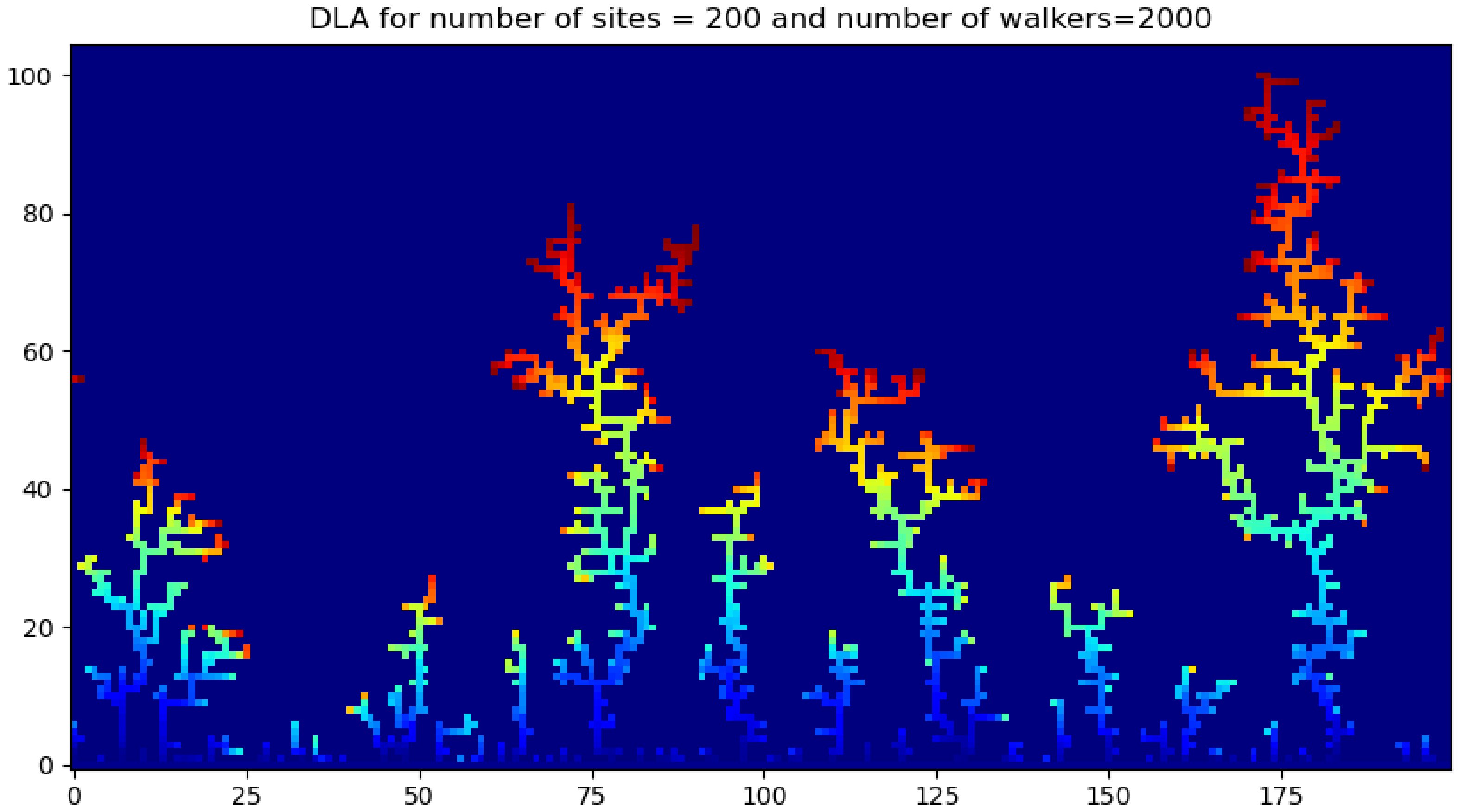
*نام: محمد جمشیدی*

*شماره دانش‌جویی: ۹۸۱۰۰۷۱۸*

## ۱.خوشته تجمع محدود (۴,۶)

در اینجا یک بذر خطی به طول ۲۰۰ داریم. سپس از ارتفاعی بالاتر از این بذر یک ولگرد را رها می‌کنیم تا هرچقدر که دلش می‌خواهد ولگردی کند! اما به محض اینکه با این بذر یا خوشه‌ای که در هر لحظه از زمان ساخته می‌شود تماس پیدا کند، ولگردی خود را متوقف می‌کند و جذب خوشه می‌شود. شرایط مرزی برای راستای افقی را تناوبی گرفته‌ام. توضیحات لازم در مورد توابعی که برای پیاده‌سازی الگوریتم استفاده کردم به صورت کامنت در داخل کد سوال موجود است. به طور کلی ابتدا یک آرایه‌ی دوبعدی را با صفر مقدار دهی اولیه می‌کنم. این آرایه قرار است همه‌ی سلول‌های خوشه را در بر بگیرد. در ادامه اولین سطر آرایه را یک می‌کنم که همان بذر خطی اولیه است. همچنین تابعی تعریف می‌کنم که در هر مرحله (منظور از مرحله یک دور کامل ولگردی است!) بیش‌ترین ارتفاع خوشه را بدست می‌آورد تا از چند خانه بالاتر از این ارتفاع بیشینه ولگرد دوبُعدیِ بَعدی را رها کنیم. در نهایت پس از آن‌که به تعداد قابل توجهی ولگرد به خوشه متصل شدند، آرایه‌ی نهایی را با یک رنگ‌نگاشت نمایش می‌دهیم. تصاویر زیر بدست می‌آید. همچنین آرایه‌ی نهایی به صورت یک فایل npy. ذخیره شده است و می‌توان آن را در کد اجرا کرد.

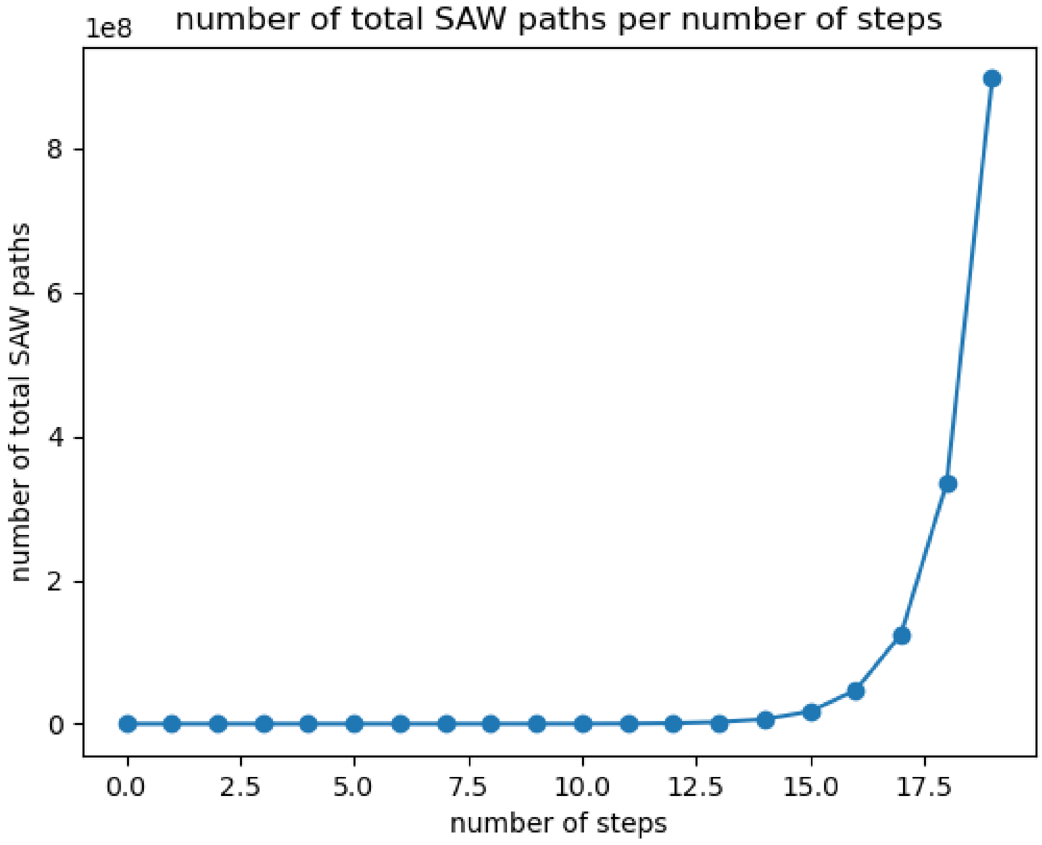





در عکس‌های بالا رفتار رقابتی به خوبی قابل مشاهده است. با افزایش تعداد ذرات خوشه‌های پر شاخ‌وبرگ‌تری مشاهده می‌شود.

## ۲.شمارش گشت‌های خود پرهیز (۴,۷)

برای پیاده سازی این مسئله از یک الگوریتم بازگشتی ساده (backtracking) استفاده کرده‌ام. در ضمن برای شمردن تعداد مسیرهای خودپرهیز از تقارن خاصی که در مورد یک لتیس دوبعدی وجود دارد استفاده کرده‌ام. در واقع تمام مسیرهای خود پرهیز به جز مسیرهایی که تماماً خط راست هستند، تقارن هشتایی دارند. یعنی همه‌ی آن‌ها صرفا دوران یافته‌ی هم‌دیگر هستند. در نتیجه کافی‌ست تعداد این مسیرهای غیرتکراری را شمارش کنیم و حاصل را در ۸ ضرب کنیم و با ٤ (٤ مسیر خودپرهیز تماماً راست داریم) جمع کنیم. برای شمارش این مسیرها، یک تابع بازگشتی تعریف می‌کنیم. این تابع بازگشتی در همه‌ی خانه‌هایی که با تعداد گام محدود در دسترس ولگرد است جستجو می‌کند و هرگاه یک مسیر خودپرهیز را تمام کرد شمارنده‌ی در نظر گرفته شده را یک واحد زیاد می‌کند. در صورتی که نتوانست مسیر خودپرهیز را انتخاب کند، آنقدر به عقب برمی‌گردد تا حق انتخاب یک مسیر متفاوت را بدست آورد و سپس مراحل گفته شده را تکرار می‌کند. توضیحات توابع ساخته شده به صورت کامنت در داخل کد موجود است. در ضمن برای اعداد بزرگ‌تر از ۲۰ زمان اجرای کد خیلی زیاد می‌شد و به همین خاطر تا حداکثر ۲۰ قدم ولگردی را بررسی کرده و نمودار آن را در زیر رسم کرده‌ام. توجه شود که الگوریتمی که به کار برده‌ایم برای اعداد بزرگ اصلاً الگوریتم بهینه‌ای نیست. ولی خب از همین تعداد محدود داده‌ای که داریم هم می‌توان نتایج لازم را استنتاج کرد.

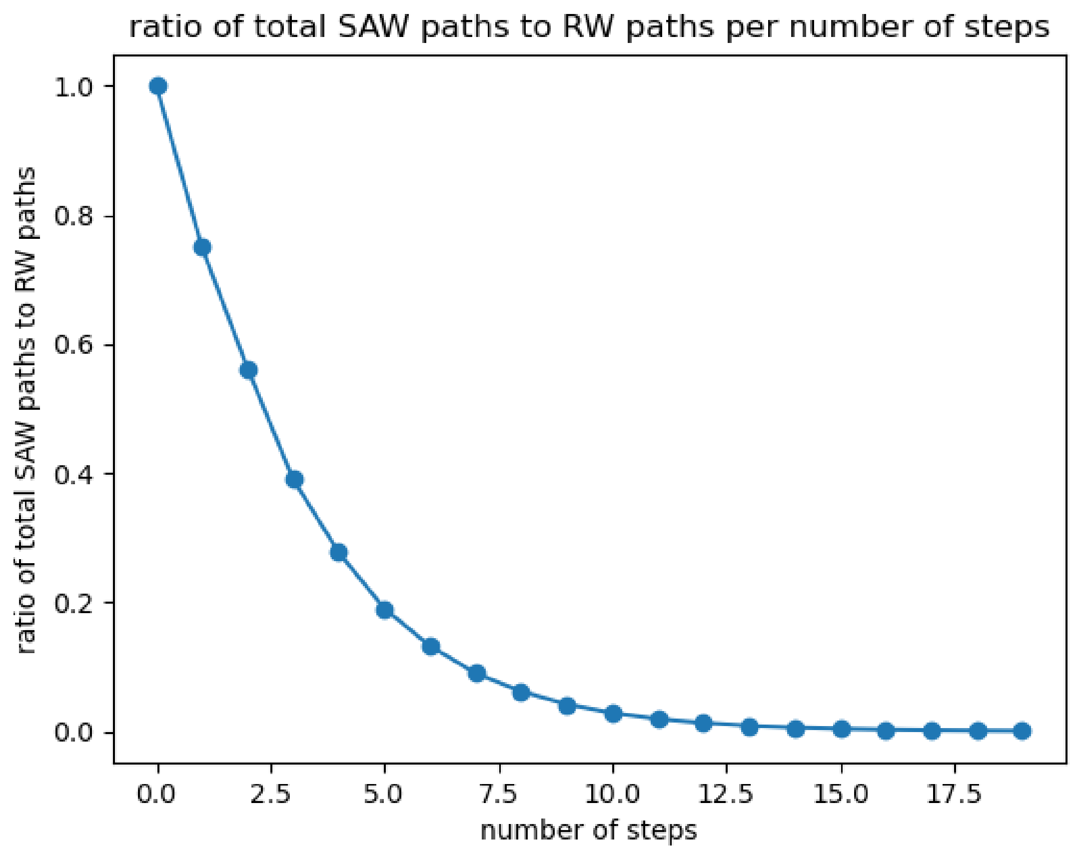



داده‌های عددی نمودار بالا در فایل total_saw_paths_per_n.npy قابل بازیابی است. در جدول زیر هم آمده است.

| | |
|---|---|
| ۱ | 4 |
| ۲ | 12 |
| ۳ | 36 |
| ۴ | 100 |
| ۵ | 284 |
| ۶ | 780 |
| ۷ | 2172 |
| ۸ | 5916 |
| ۹ | 16268 |
| ۱۰ | 44100 |

| | |
|---|---|
| ۱۱ | 120292 |
| ۱۲ | 324932 |
| ۱۳ | 881500 |
| ۱۴ | 2374444 |
| ۱۵ | 6416596 |
| ۱۶ | 17245332 |
| ۱۷ | 46466676 |
| ۱۸ | 124658732 |
| ۱۹ | 335116620 |
| ۲۰ | 897697164 |

ستون سمت چپ تعداد گام‌های ولگردی مجاز و ستون سمت راست تعداد مسیرهای خودپرهیز است.

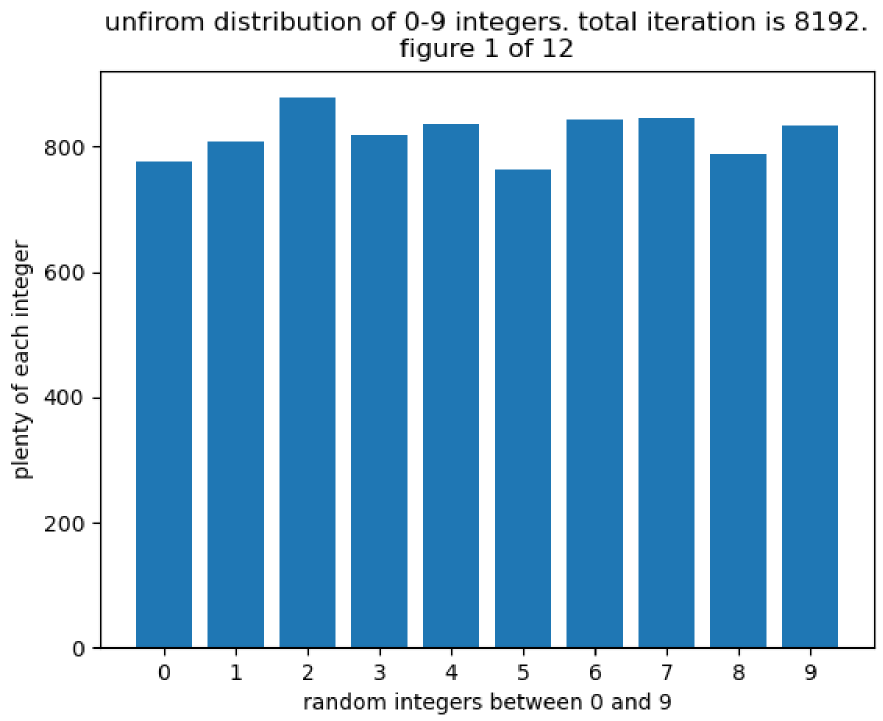


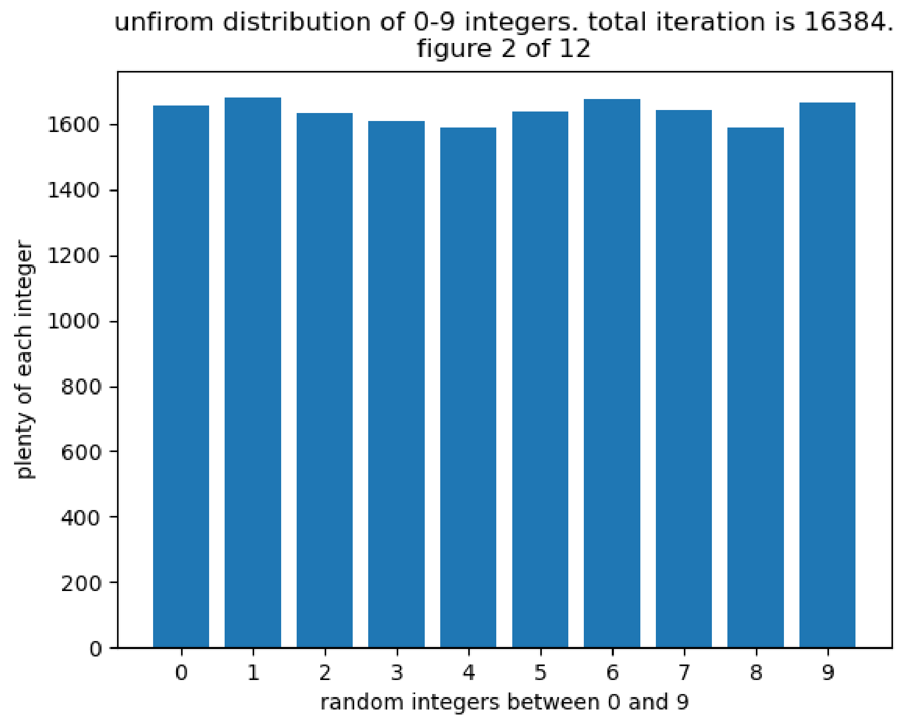

همانطور که در نمودار دوم دیده می‌شود، با افزایش تعداد قدم‌های مجاز ولگرد، تعداد نسبی مسیرهای خودپرهیز به صفر میل می‌کند. و این یعنی برای تعداد قدم‌های بزرگ احتمال آن که ولگرد به خودش برخورد کرده باشد بیش‌تر است.

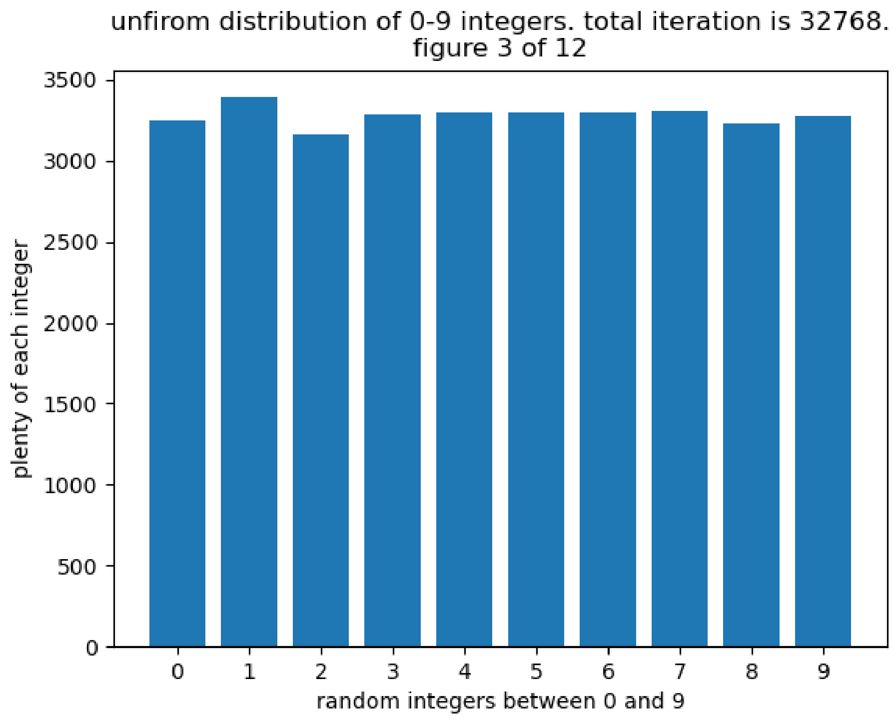
## ۳.تست RNG (۶,۱)

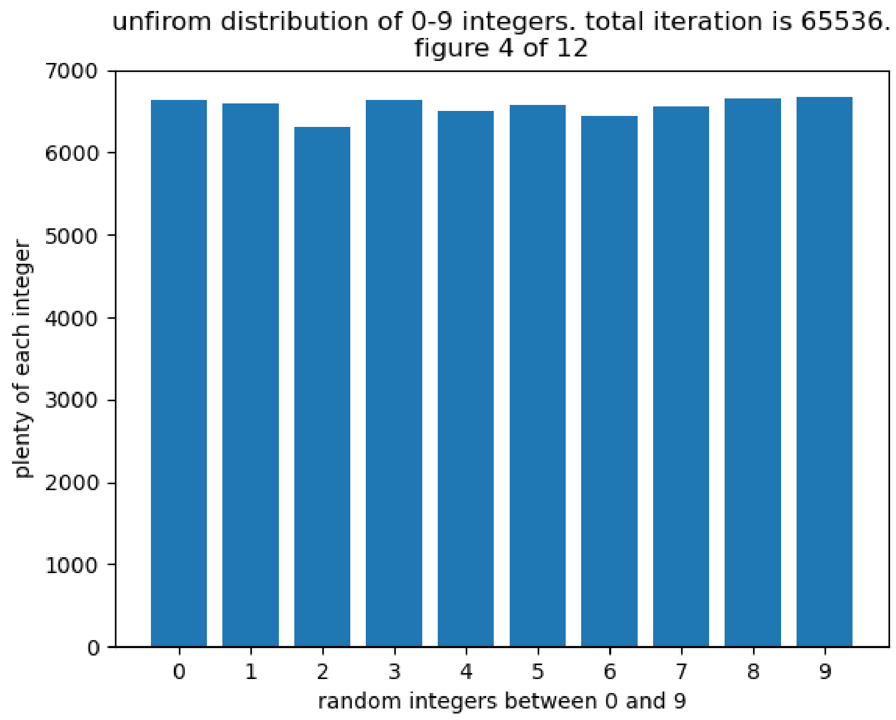
در اینجا می‌خواهیم بررسی کنیم تابع مولد numpy.random.randint() که در مسئله‌های قبلی به کرات از آن استفاده کرده‌ایم، چه میزان واقعاً یک توزیع رندوم را تولید می‌کند. کاری که انجام می‌دهیم این است که به تعداد زیادی با فراخوانی تابع یادشده، اعداد رندوم از ۰ تا ۹ تولید می‌کنیم. در نهایت توزیع این اعداد را به صورت یک هیستوگرام رسم می‌کنیم. اگر توزیع ما واقعاً رندوم و یکنواخت باشد، باید در تعداد تکرارهای بالا، مشاهده کنیم که همه‌ی اعداد تقریباً به یک اندازه تکرار شده است. همان‌طور که از نمودارهای زیر مشاهده می‌شود، در تعداد تکرارهای خیلی بالا، افت و خیز بسیار ناچیز می‌شود.

گام‌های تکرارها را به صورت [2^x for x in range(13, 25)] گرفته‌ام. فایده‌ی این کار آن است که باعث می‌شود در نهایت نمودار لگ-لگ سیگما بر حسب تعداد گام به صورت یک نمودار یکنواخت باشد. اگر توزیع گام‌ها را یکنواخت می‌گرفتیم، در این صورت نقاط برای اعداد بزرگ‌تر فشرده‌تر می‌شدند. نمودارهای زیر بدست آمده است.

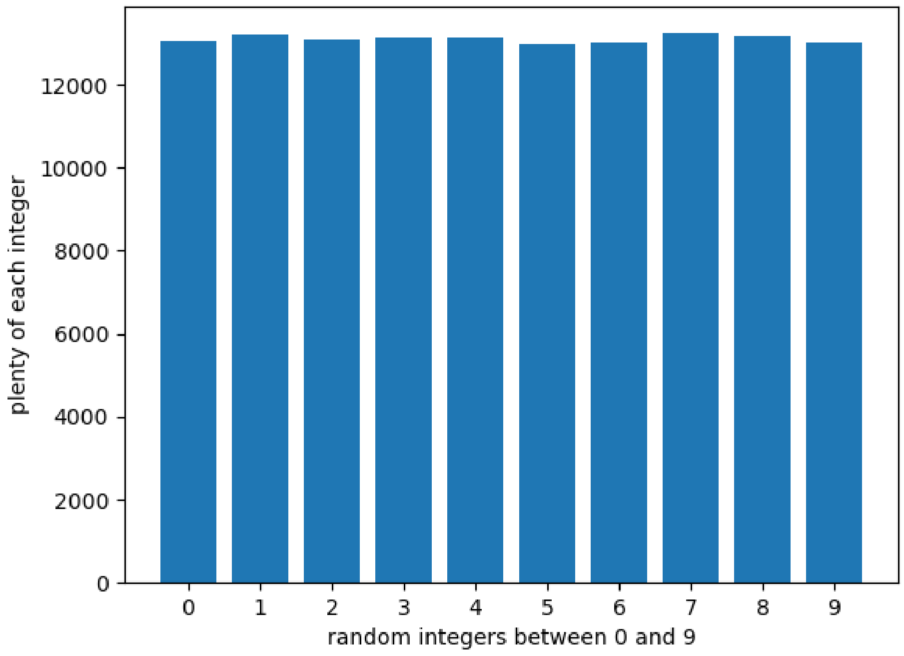


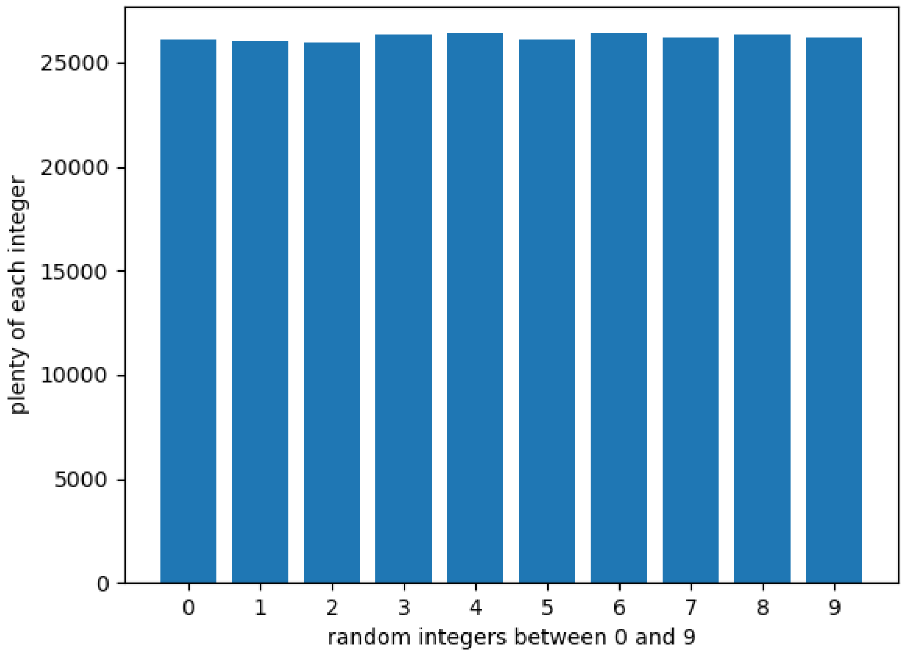



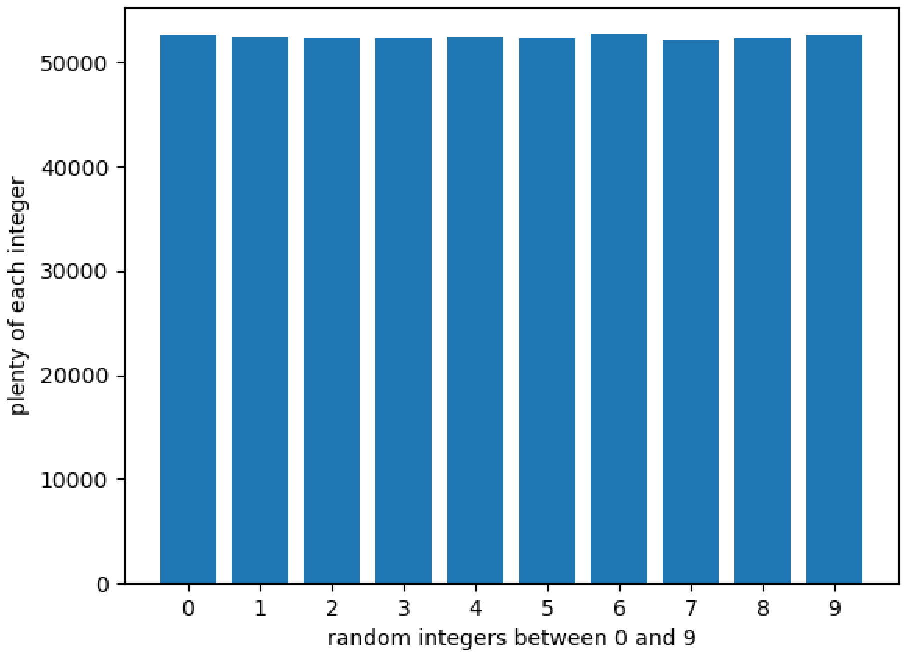
unfirom distribution of 0-9 integers. total iteration is 131072.
figure 5 of 12



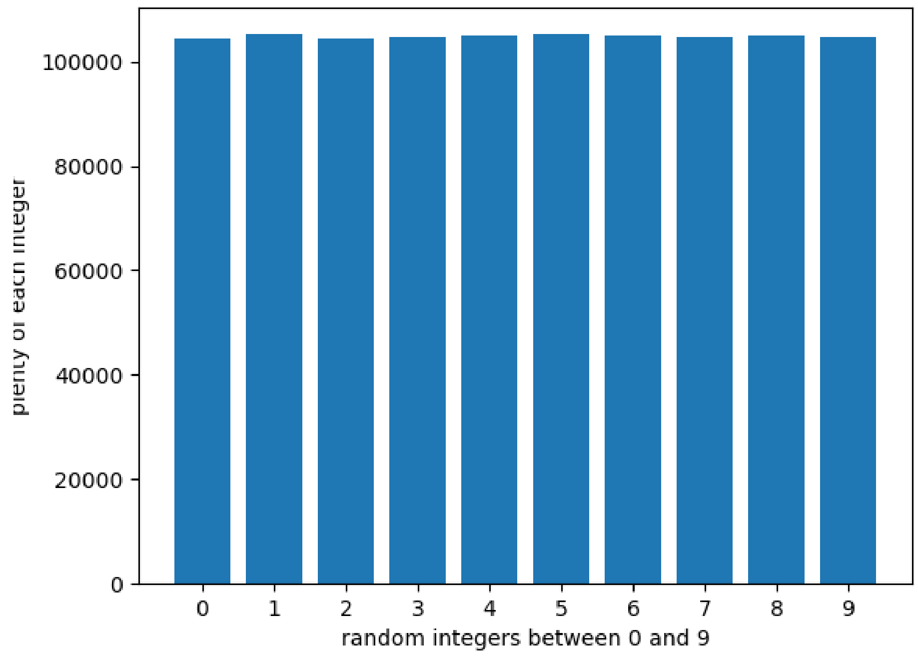

unfirom distribution of 0-9 integers. total iteration is 262144.
figure 6 of 12



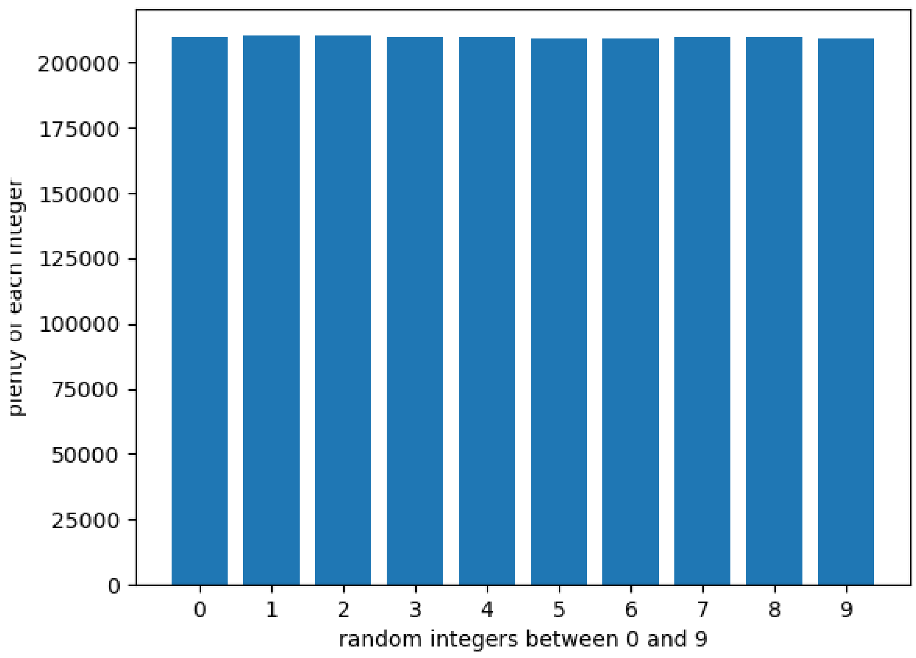

unfirom distribution of 0-9 integers. total iteration is 524288.
figure 7 of 12



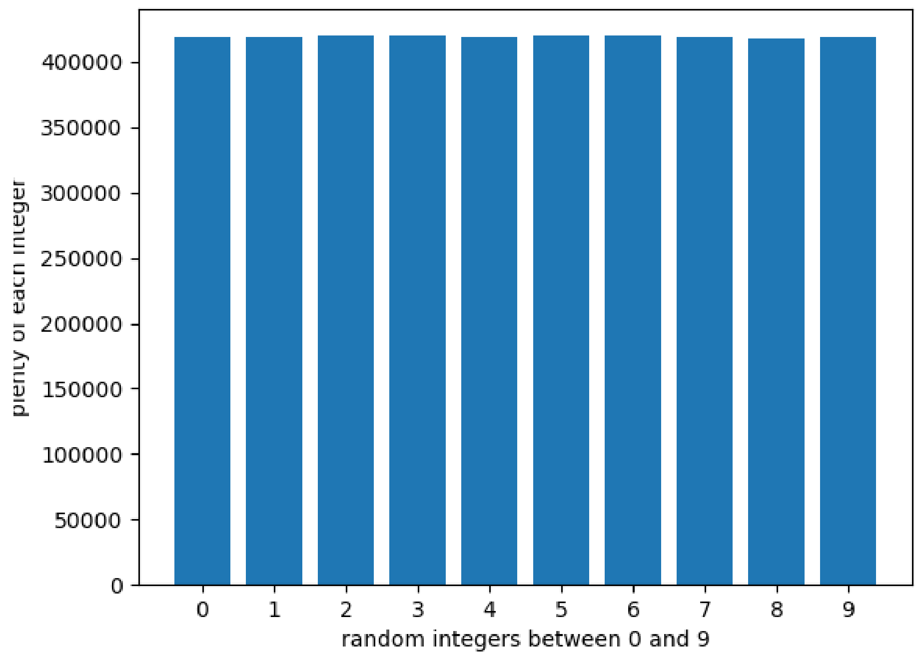

unfirom distribution of 0-9 integers. total iteration is 1048576.
figure 8 of 12

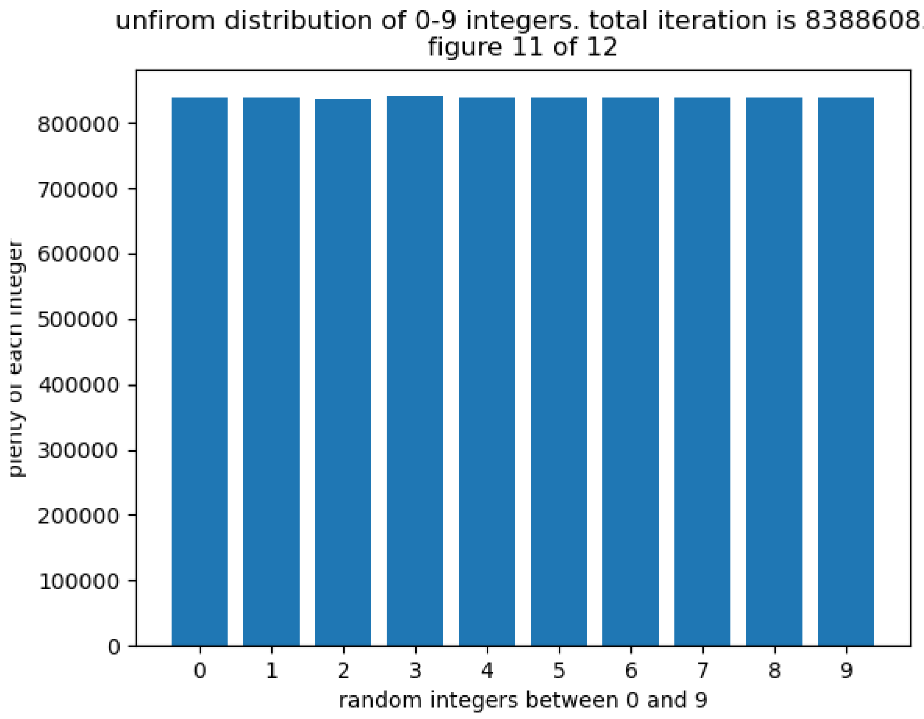


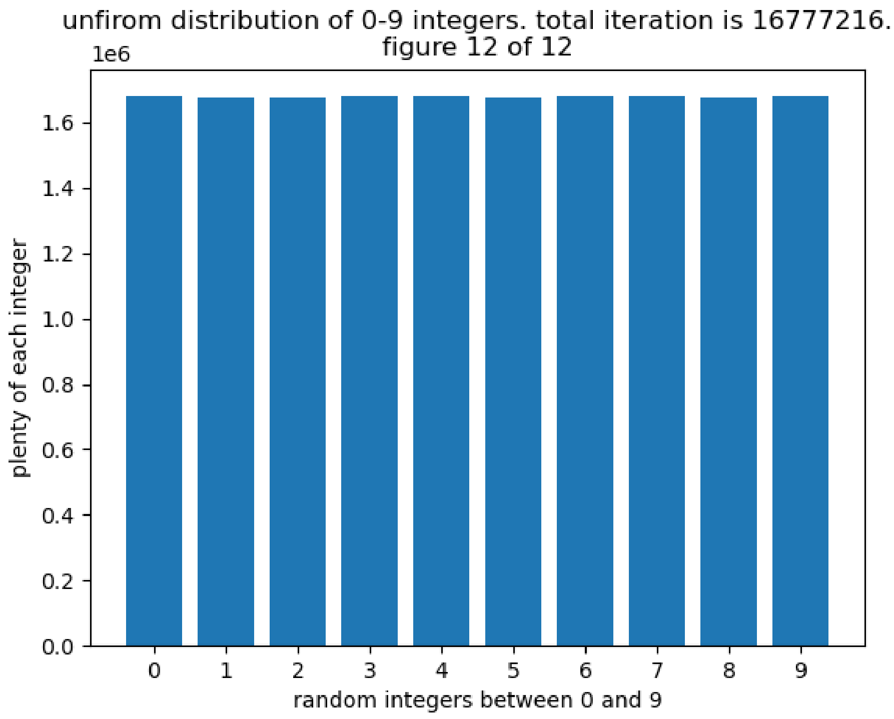

unfirom distribution of 0-9 integers. total iteration is 2097152.
figure 9 of 12




unfirom distribution of 0-9 integers. total iteration is 4194304.
figure 10 of 12



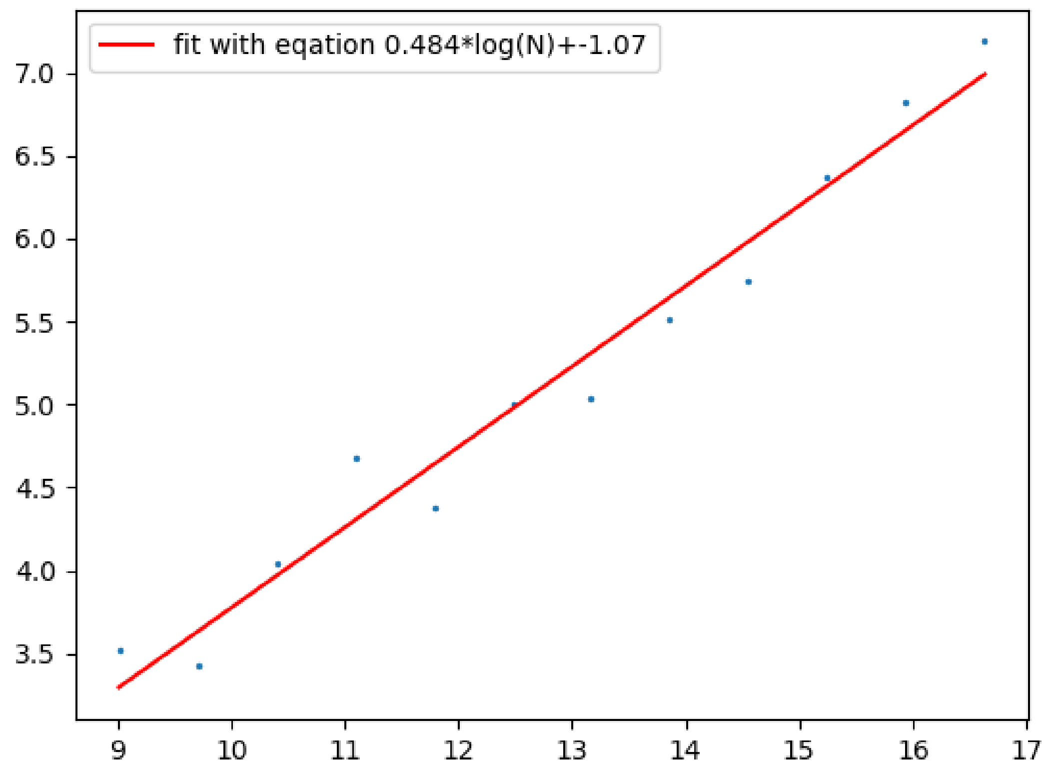



همان طور که مشاهده می‌شود، افت و خیز اعداد رندوم با افزایش تعداد تکرار، کاهش می‌یابد. در ادامه انحراف از معیار را به صورت تابعی از تعداد تکرارها رسم می‌کنم. از منظر تئوری روابط زیر را داریم.

$$\frac{\sigma}{N} \sim \frac{1}{\sqrt{N}} \to \log \sigma \sim 0.5 \log N$$

پس انتظار داریم در نمودار لگ–لگ خود یک خط با شیب حدوداً نیم مشاهده کنیم. نمودار به صورت زیر است و با تئوری کاملاً مطابقت دارد.

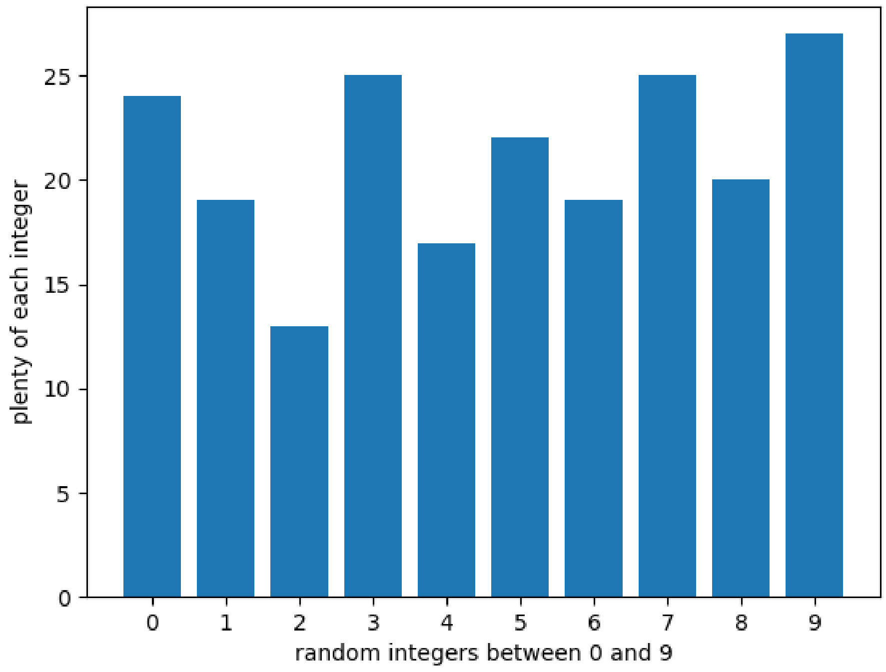



البته مقداری افت و خیز نسبت به خط بهینه مشاهده می‌شود. این افت و خیزها ریشه در همان رندوم بودن اعداد دارد. در مقایسه‌ی این تمرین با ول‌نشست باید گفت آنجا هم دقیقاً ذرات به صورت رندوم نشست می‌کردند و در نتیجه لایه‌ی ما در واقع یک هسیتوگرام بود. به همین خاط هیستوگرام‌های این مسئله کاملاً شبیه به لایه نشانی در ول‌نشست است.

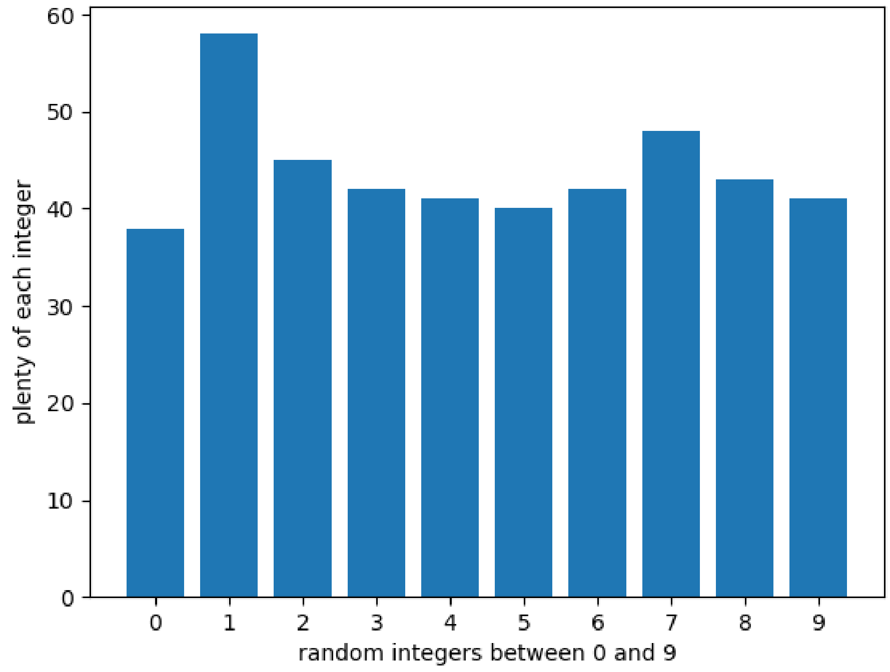
## ۴. بررسی همبستگی اعداد تصادفی (۶,۲)

در اینجا قصد داریم بررسی کنیم آیا اعدادی که توسط تابع مولد numpy.random.randint() تولید می‌شوند همبسته هستند یا خیر. برای این منظور رشته‌ای از اعداد رندوم تولید می‌کنیم و در این رشته هر عددی که قبل از ٤ آمده است را در یک آرایه ذخیره می‌کنیم. این کار را به تعداد زیادی تکرار می‌کنیم و در نهایت توزیع آماری اعداد را به صورت هیستوگرام رسم می‌کنیم. توضیحات مربوط به توابع در داخل کد داده شده است. نمودارها در زیر آورده شده است.

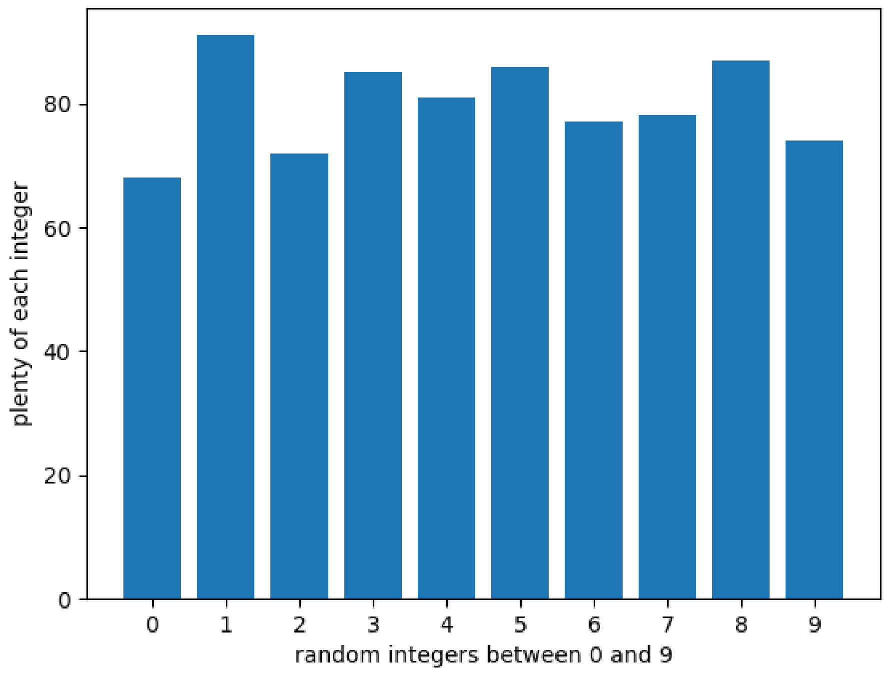
unfirom distribution of 0-9 integers. total iteration is 2048.
figure 1 of 14



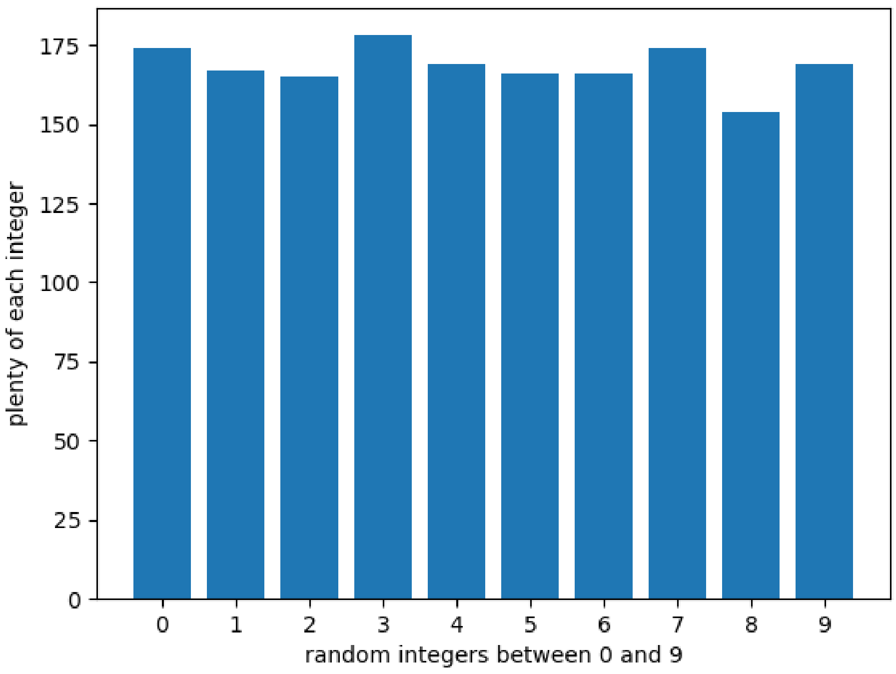

unfirom distribution of 0-9 integers. total iteration is 4096.
figure 2 of 14



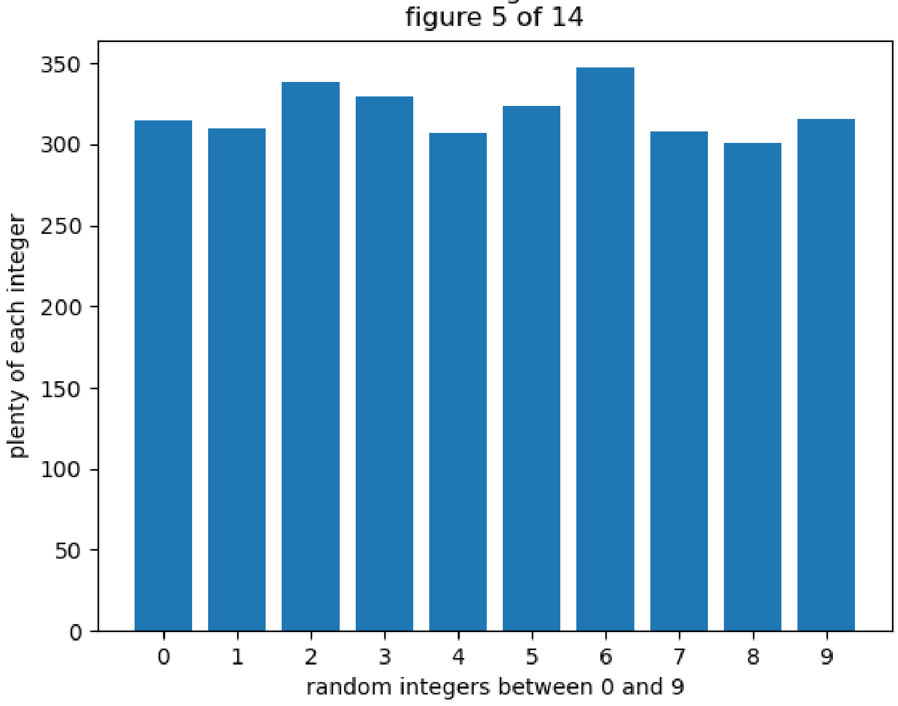

unfirom distribution of 0-9 integers. total iteration is 8192.
figure 3 of 14



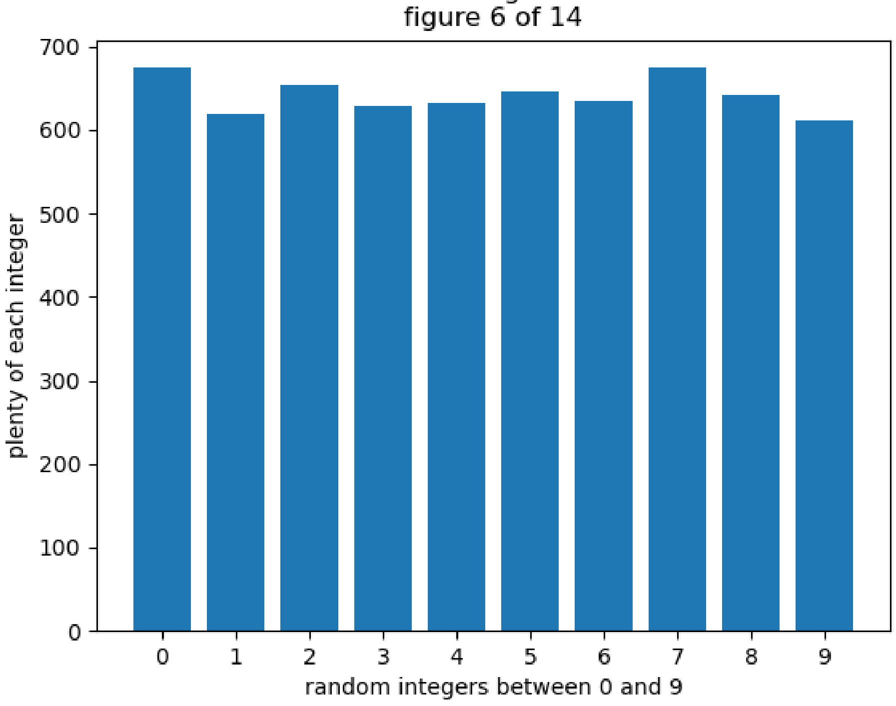

unfirom distribution of 0-9 integers. total iteration is 16384.
figure 4 of 14

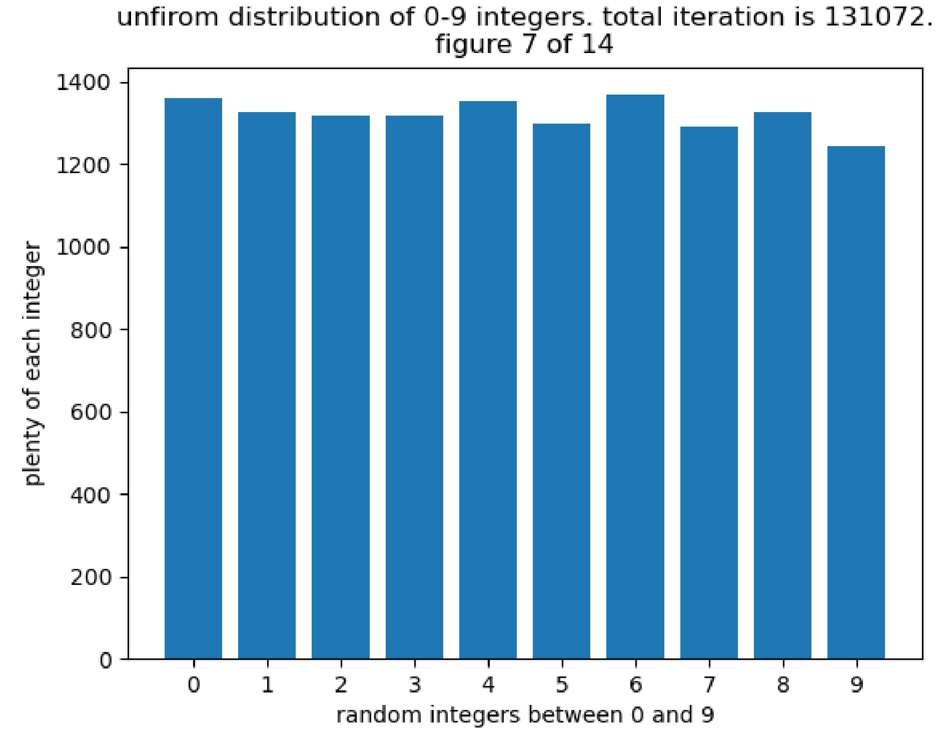


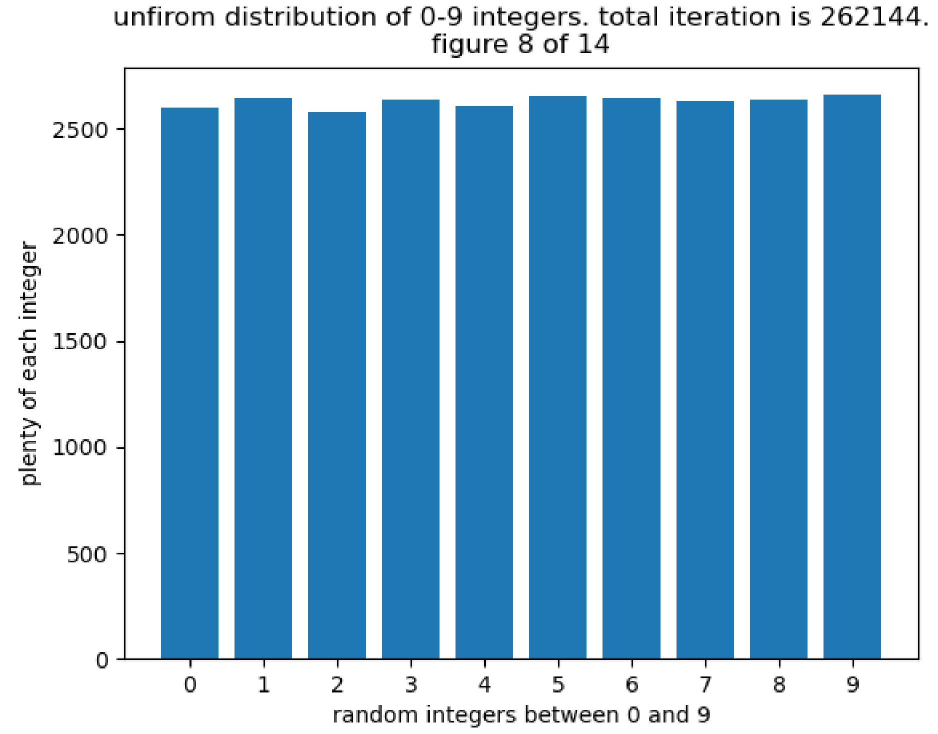

unfirom distribution of 0-9 integers. total iteration is 32768.
figure 5 of 14

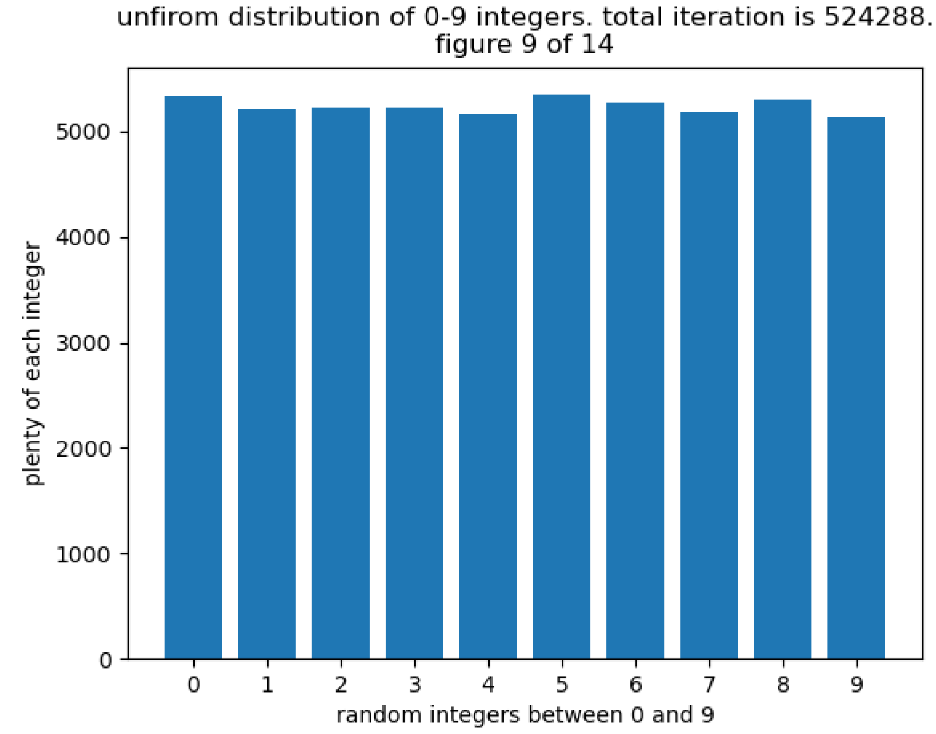


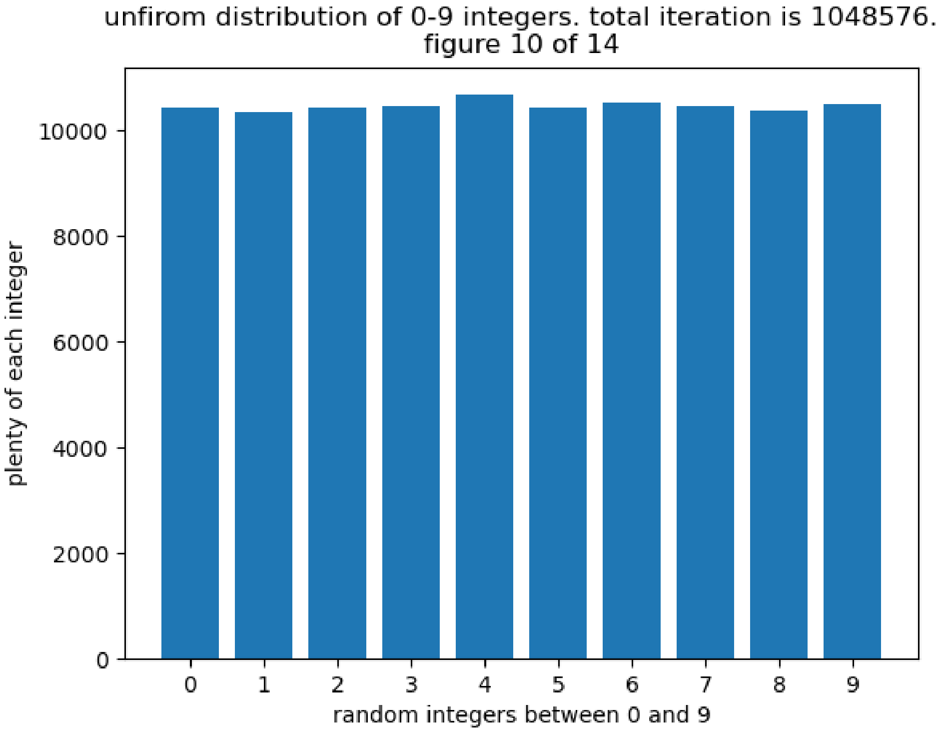

unfirom distribution of 0-9 integers. total iteration is 65536.
figure 6 of 14

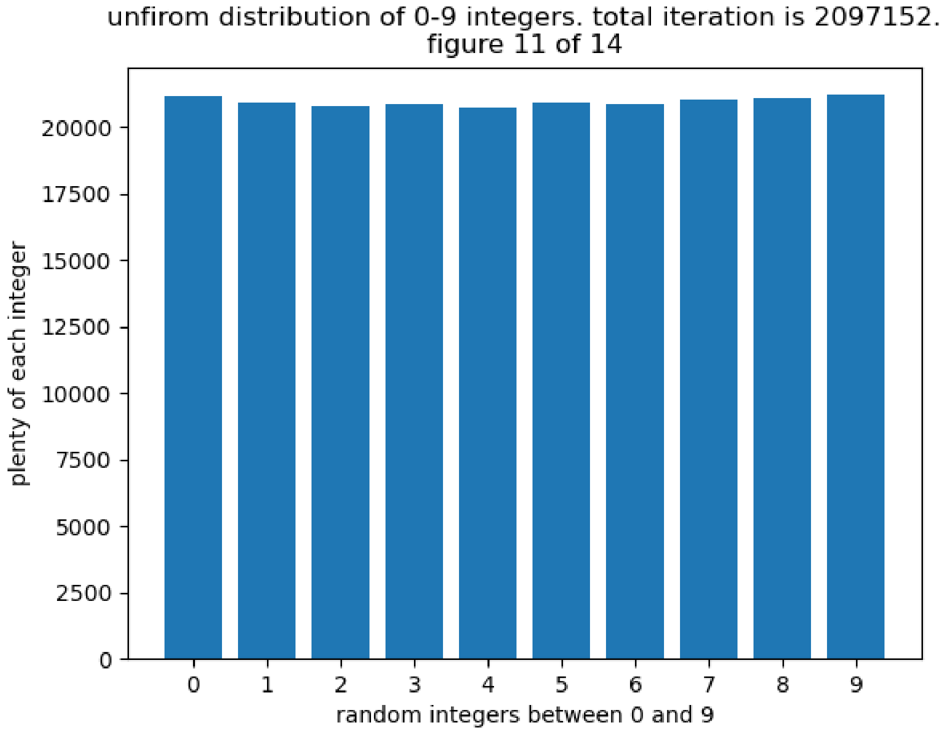


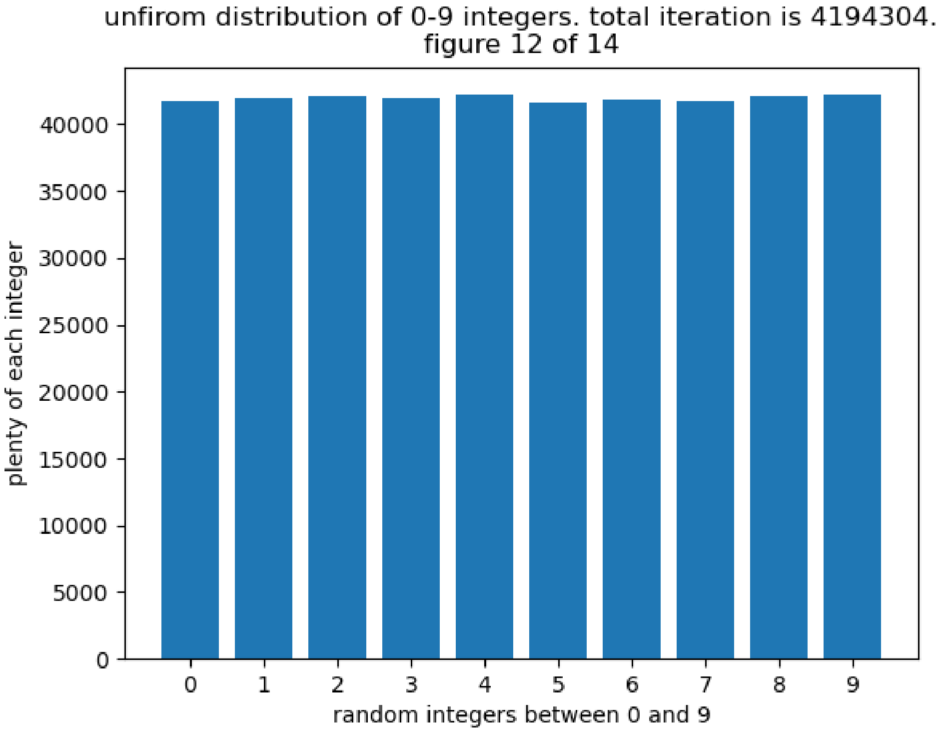

unfirom distribution of 0-9 integers. total iteration is 131072.
figure 7 of 14
plenty of each integer
random integers between 0 and 9

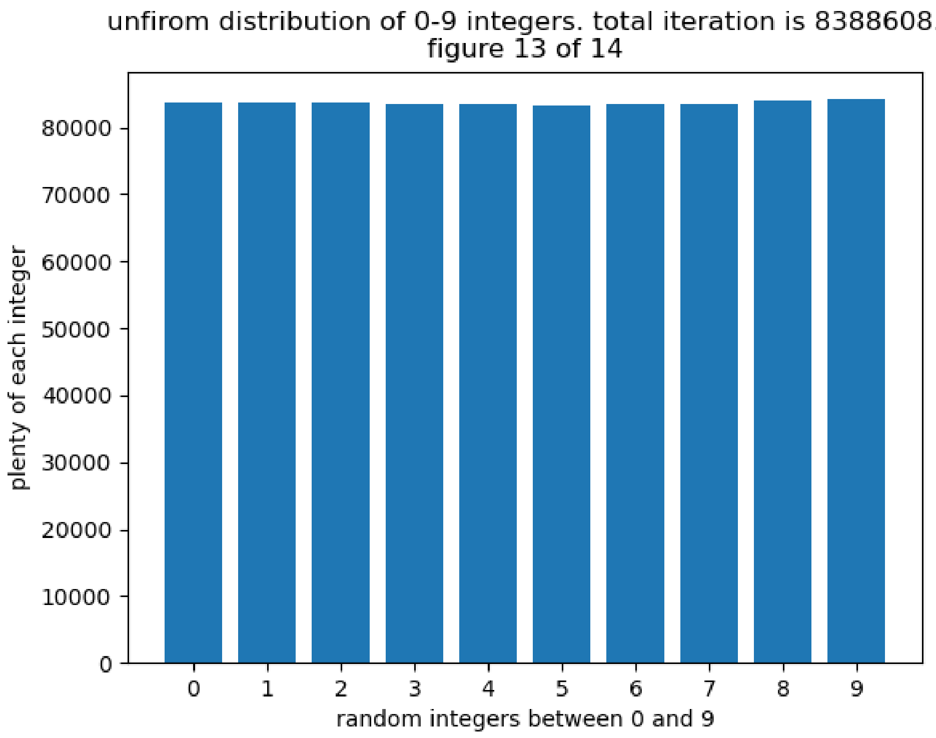

unfirom distribution of 0-9 integers. total iteration is 262144.
figure 8 of 14
plenty of each integer
random integers between 0 and 9

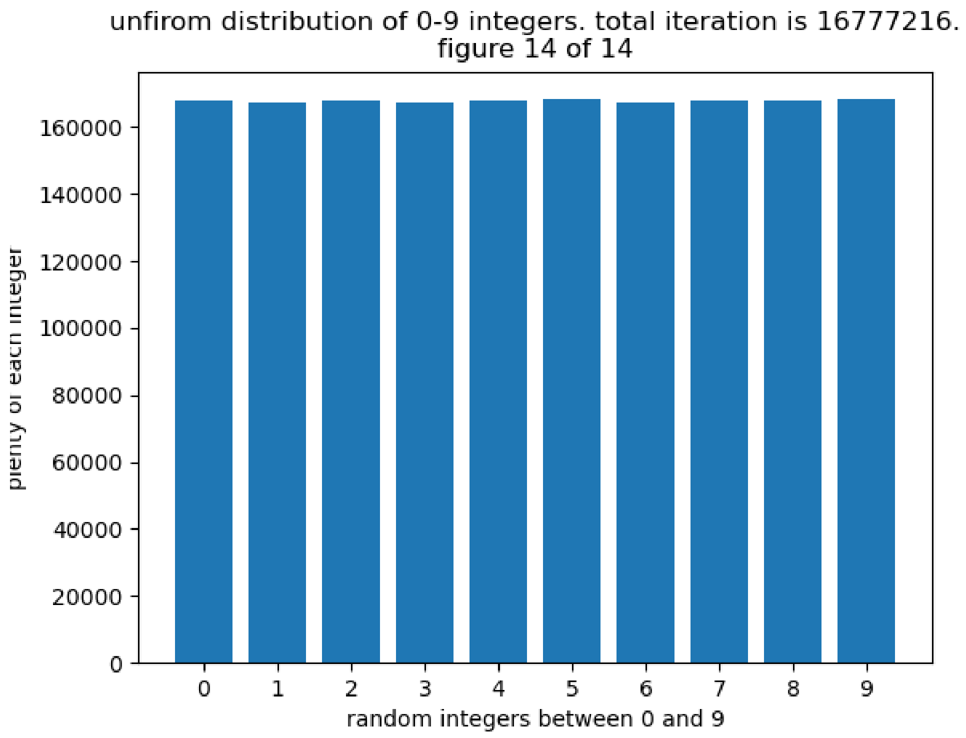

unfirom distribution of 0-9 integers. total iteration is 524288.
figure 9 of 14
plenty of each integer
random integers between 0 and 9


unfirom distribution of 0-9 integers. total iteration is 1048576.
figure 10 of 14
plenty of each integer
random integers between 0 and 9

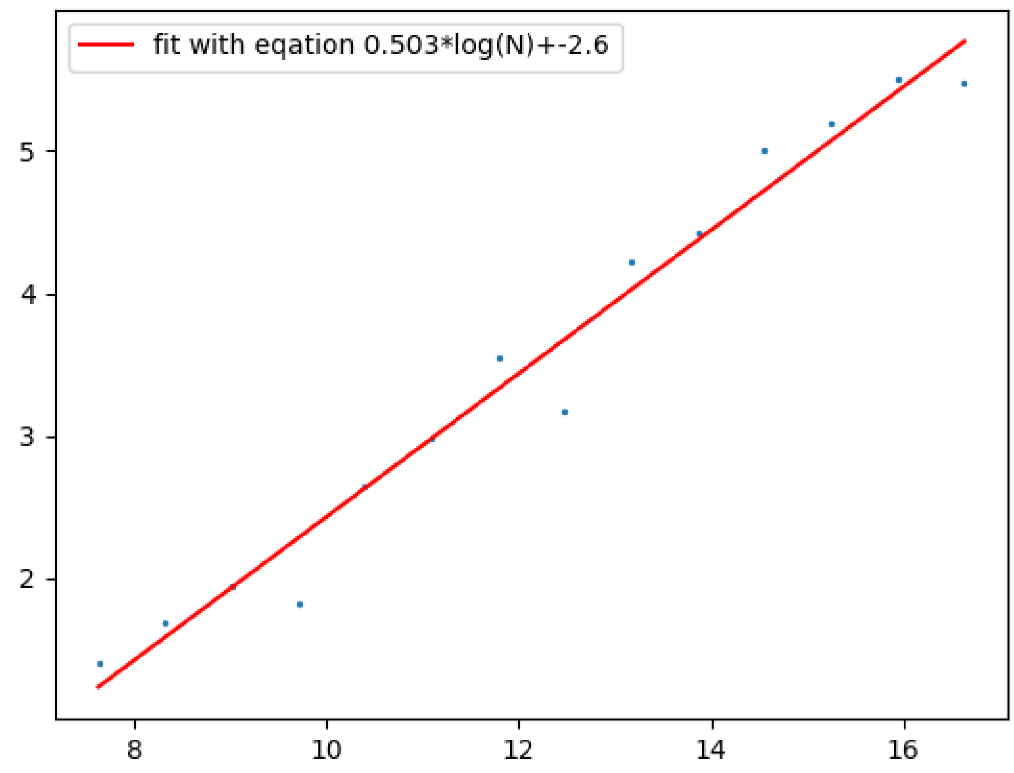

unfirom distribution of 0-9 integers. total iteration is 2097152.
figure 11 of 14
plenty of each integer
random integers between 0 and 9


unfirom distribution of 0-9 integers. total iteration is 4194304.
figure 12 of 14
plenty of each integer
random integers between 0 and 9




تئوری که در اینجا داریم با تئوری مسئله‌ی قبل یکسان است. بنابراین کافی‌ست نمودار لگ-لگ انحراف از معیار را برحسب تعداد آزمایش‌ها رسم کنیم. انتظار داریم نمودار ما یک خط با شیب ۰٫۵ باشد. نمودار به صورت زیر بدست آمده است و بنابراین با تئوری منطبق است.

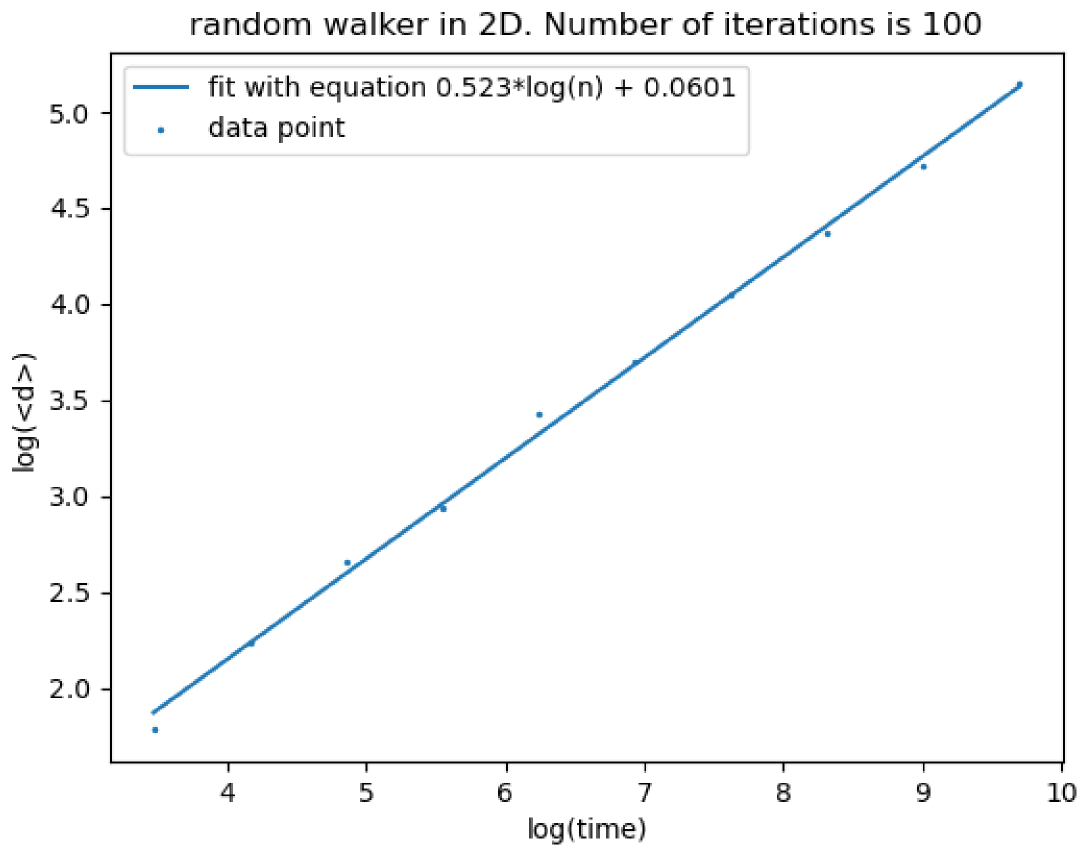


## ۵.ولگرد دو بعدی (۴,۵)

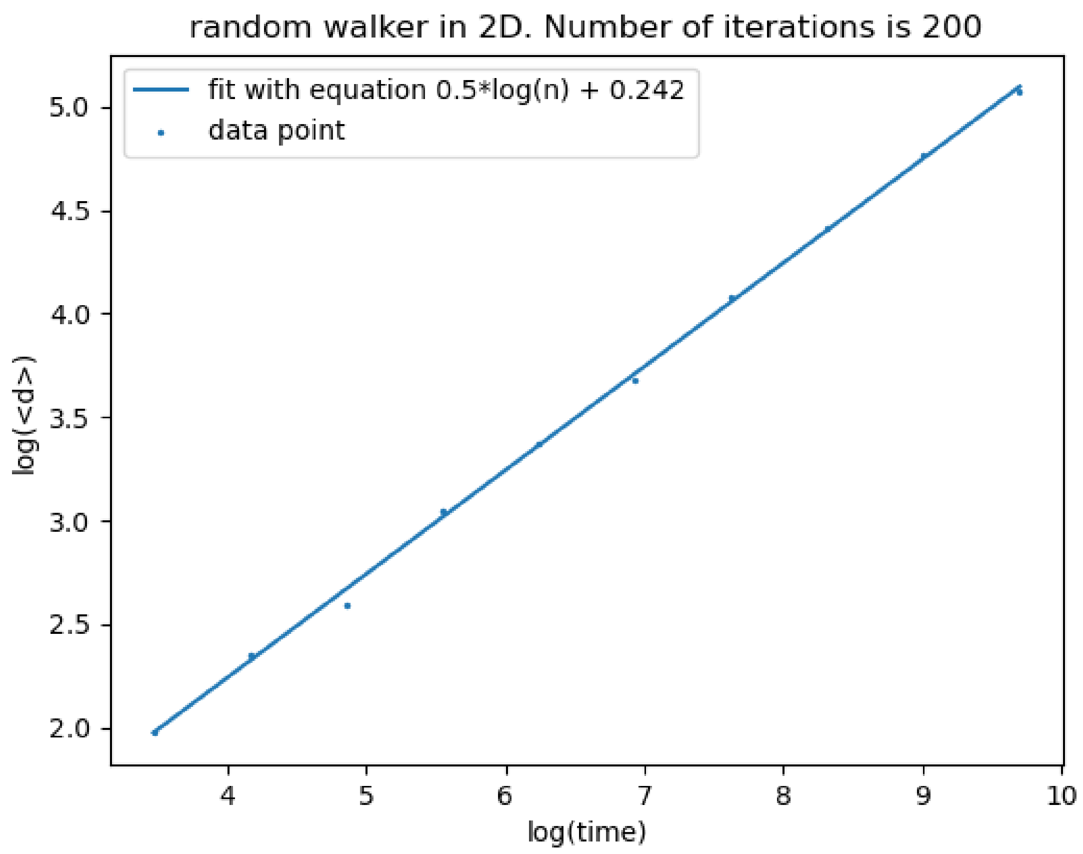
در اینجا یک ولگشت دوبعدی را بر روی یک شبکه مربعی و از مبدا رها می‌کنیم تا برای خود ولگردی کند. تعداد گام‌هایی که ولگرد مجاز است بردارد را خودمان تعیین می‌کنیم. این تعداد گام‌ها در واقع همان کمیت زمان در مسئله‌ی اصلی است. برای این منظور هربار دو متغیر x, y را که با صفر مقداردهی اولیه شده‌اند، انتخاب می‌کنیم و به صورت رندوم مقدارشان را یک واحد زیاد یا کم می‌کنیم. در نهایت جذر مجموع مجذور این دو متغیر، فاصله‌ی ولگرد تا مبدا است. اینکار را به تعداد زیادی و برای زمان‌های مختلف انجام می‌دهیم. در نهایت نمودار لگ-لگ میانگین فاصله از مبدا را برحسب زمان یا همان تعداد گام‌های مجاز رسم می‌کنیم. شیب این نمودار کمیت مهمی است چون:

$$< r^2 > = 2dDt \rightarrow \sqrt{< r^2 >} \sim t^{0.5} \rightarrow \log\left(\sqrt{< r^2 >}\right) \sim 0.5logt + C$$

بنابراین به لحاظ تئوری پیش‌بینی‌مان این است که شیب خط نمودارمان باید به مقدار ۰,۵ نزدیک باشد. نمودارهای زیر موید این موضوع است.




اگر تعداد تکرار را دوبرابر کنیم، نتیجه خیلی بهتر از این‌ها هم می‌شود.



به این ترتیب صحت رابطه‌ی ۱۲ کتاب تأیید می‌شود. ☺