

تمرین شبیه سازی سری سوم  
علی قبله ۹۹۱۰۹۹۷۱

سوال ۱: پایین نشست

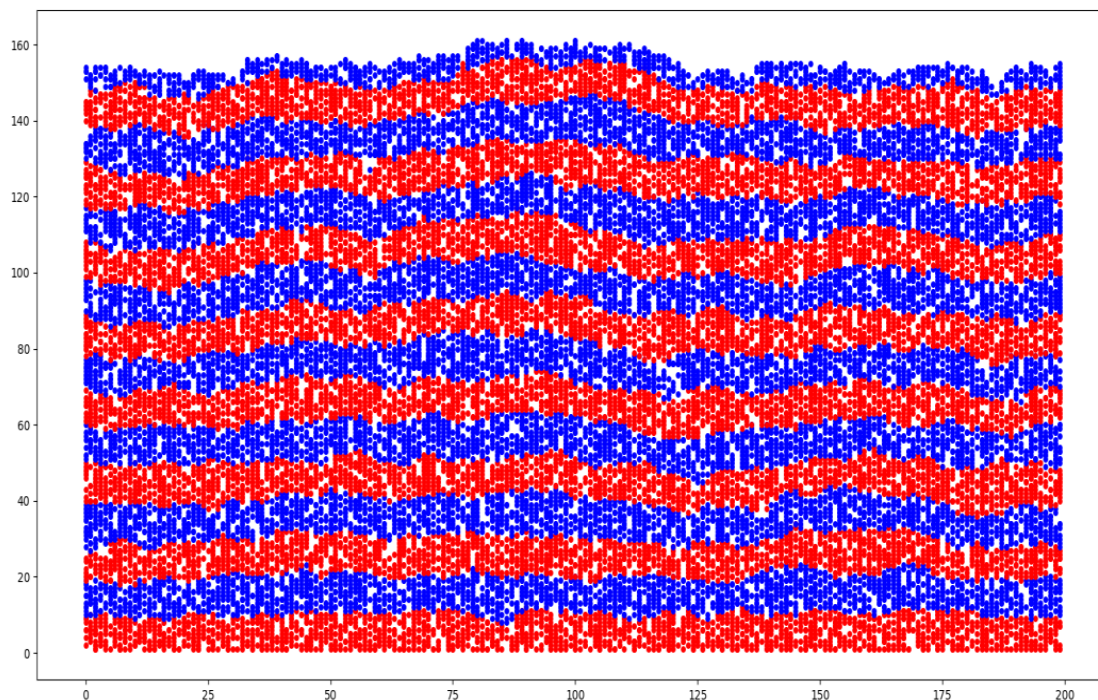
در مدل پایین نشست، یک ذره به نوعی یک خانه در چپ یا راست خود را مشاهده می کند و هنگام سقوط می تواند با توجه به این که کدام ستون از خود ستون رندوم انتخاب شده یا دو همسایهی چپ و راست آن کوتاهتر است در آن فرود آید. در ابتدا بایستی چند پارامتر را مشخص کنیم:

- I. طول سیستم
- II. تعداد دفعات اجرا
- III. تعداد داده های اندازه گیری شده
- IV. تعداد نقاط تا زمان اندازه گیری که پس از هر اندازه گیری ۲ برابر می شود

برای استفاده راحت تر از این متغیرها در توابع نوشته شده در کد، این متغیرها را به صورت Global تعریف کردیم.

مهمترین بخش کد که محاسبات تمام توابع تعریف شده در آن رخ میدهد، تابع `Evo` است. در ابتدا لازم است که مشخص کنیم برای مشاهده گرافیک پایین نشست کافی است که دو خط رنگ آمیزی و گرافیک را از حالت کامنت خارج کرده و خط `plt.error` را وارد کامنت کنیم تا گرافیک را مشاهده کنیم و برای مشاهده نمودار `error` `bar`، این دو دستور را خاموش و خط `plt.error` را از کامنت خارج می کنیم تا بتوانیم نمودارها را مشاهده کنیم.

حال برای ۳۱۰۰۰ نقطه در طول ۲۰۰ شکل به صورت زیر خواهد بود:



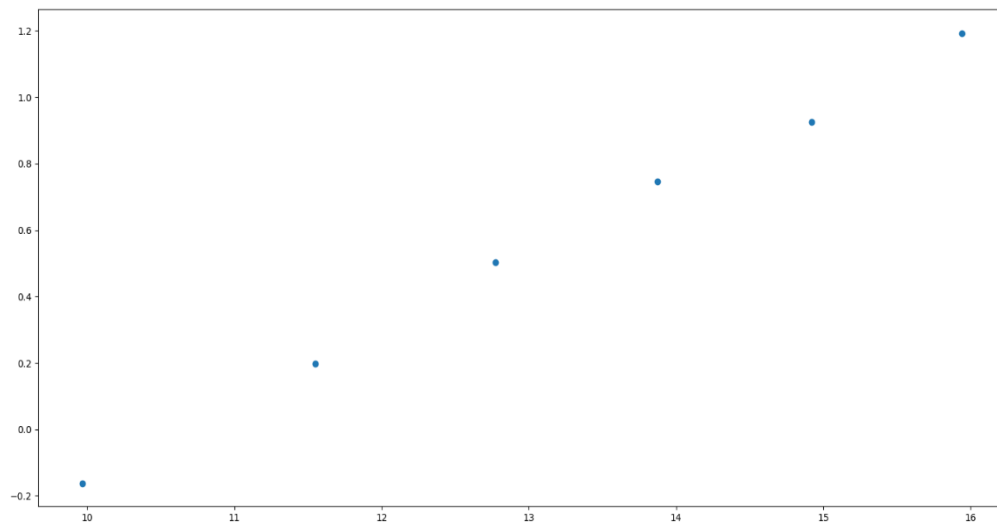
برای میانگین ارتفاع لایه ها طبق زمان (بازه های زمانی توزیع لوگاریتمی دارند) داریم:

$[0, \dots e+00$	$1.5000e+01$	$3.5000e+01$	$7.5000e+01$
$1.5500e+02$	$3.1500e+02$	$6.3500e+02$	$1.2750e+03$
$2.5550e+03$	$5.1150e+03$	$1.0235e+04$	$2.0475e+04$
$4.0955e+04$	$8.1915e+04$		

برای بتا پس از اجرای برنامه برای ۶۳۰۰۰ نقطه و ۱۵ بار خواهیم داشت:

$$\text{Beta} = 0,22$$

برای نمودار  $\log(w)$  به  $\log(t)$  داریم:



بهترین Z که بدست می آید ( با احتساب لوگاریتم طول های ۲۰۰ تا ۴۰۰ و لوگاریتم t ):

$$Z = 2,49$$

در انتها برای alpha خواهیم داشت:

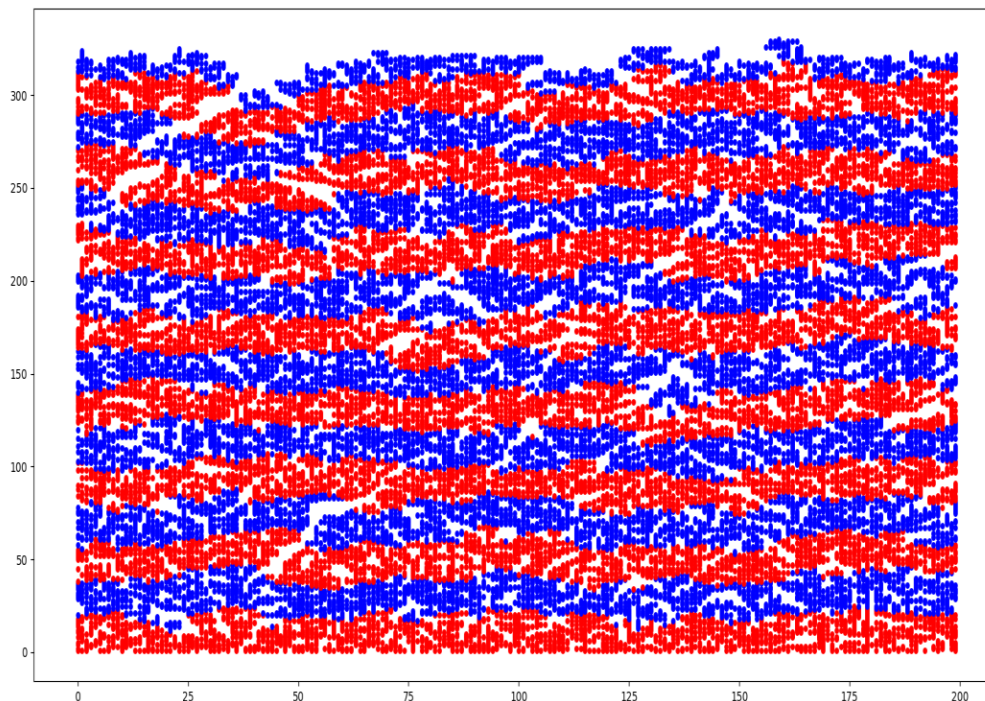
$$\text{Alpha} = Z * \text{Beta}$$

$$\text{Alpha} = 2,49 * 0,22 = 0,54$$

سوال ۲: کنار نشست

در مدل کنارنشست ذره هنگام سقوط در همان ستون اولیه سقوط میکند ولی در ارتفاعی میایستد که بلندترین همسایه (از جمله خود این ستون) قرار دارد. با این کار ستونها به جای این که کاملاً پر باشند متخلخل خواهند بود. جزئیات این سوال نیز دقیقاً همانند سوال قبل می باشد. پارامترهای تعیین شده و حضور متغیرهای Global در کنار حذف یا اضافه دستورهای Color و Graphic، همانند سوال پیشین است.

برای طول ۲۰۰ و تعداد ۳۱۰۰۰ نقطه داریم: (عکس در فایل ورژنی است که زمان بسیار کمی طول کشیده)



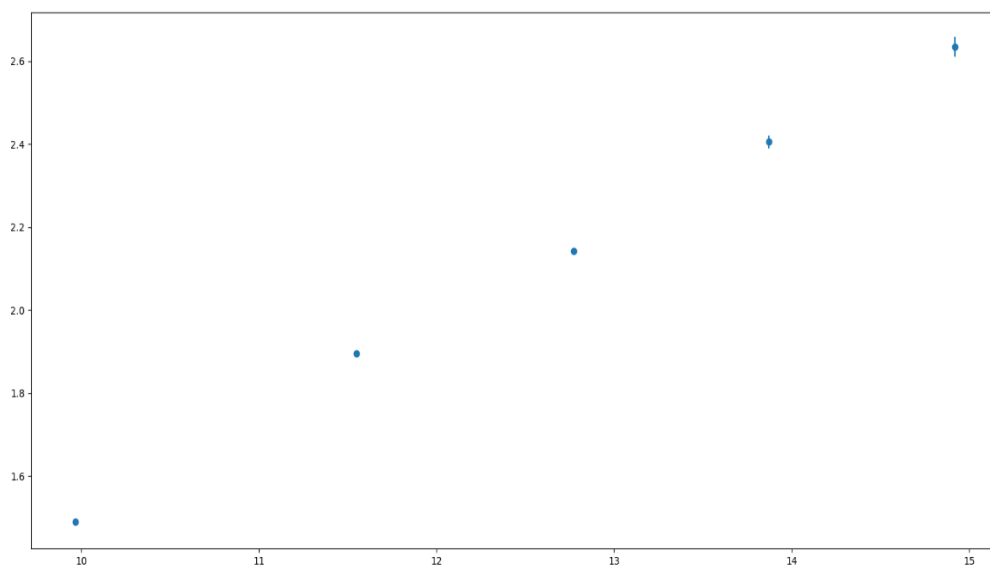
برای میانگین ارتفاع نسبت به تابع زمان داریم:

[۸,۵۲۵۰۰۰e+۰۰, ۲,۸۱۷۵۰۰e+۰۱, ۷,۰۳۵۵۰۰e+۰۱, ۱,۵۱۹۱۰۰e+۰۲,  
۳,۲۲۱۰۰۰e+۰۲, ۶,۶۶۳۱۵۰e+۰۲, ۱,۳۴۱۶۷۵e+۰۳, ۲,۷۰۷۸۶۰e+۰۳,  
۵,۴۲۹۹۸۵e+۰۳, ۱,۰۸۷۱۷۳e+۰۴, ۲,۱۷۹۰۱۵۰۰e+۰۴, ۴,۳۶۵۹۷۱۰۰e+۰۴,  
۸,۷۲۵۸۲۶۵۰e+۰۴, ۱,۷۴۶۴۹۲۸۵e+۰۵]

همانند قبل برای بتا نیز خواهیم داشت:

Beta = ۰,۲۲

نمودار  $\log(w)$  به  $\log(t)$ :



همانند بخش قبل برای  $Z$  خواهیم داشت:

$$Z = 3,3$$

و برای  $\alpha$  داریم:

$$\alpha = 0,76$$

سوال ۳:

برای این سوال فرصت نشد کد بزنم اما:

ایده اصلی سوال کافی است یک نقطه را به عنوان رندوم در صفحه در پایین ترین خط انتخاب و با استفاده از رندوم هر نقطه جدیدی را ابتدا چک کنیم که در همسایگی نقطه ای که پر است قرار دارد یا خیر! اگر جواب خیر بود نقطه از سیستم حذف می شود اما اگر پاسخ بله بود در آن نقطه به نقطه پیشین می چسبد. این تنها شرطی است که لازم است به برنامه اضافه کنیم و کد سوال قبل را دوباره تکرار می کنیم.

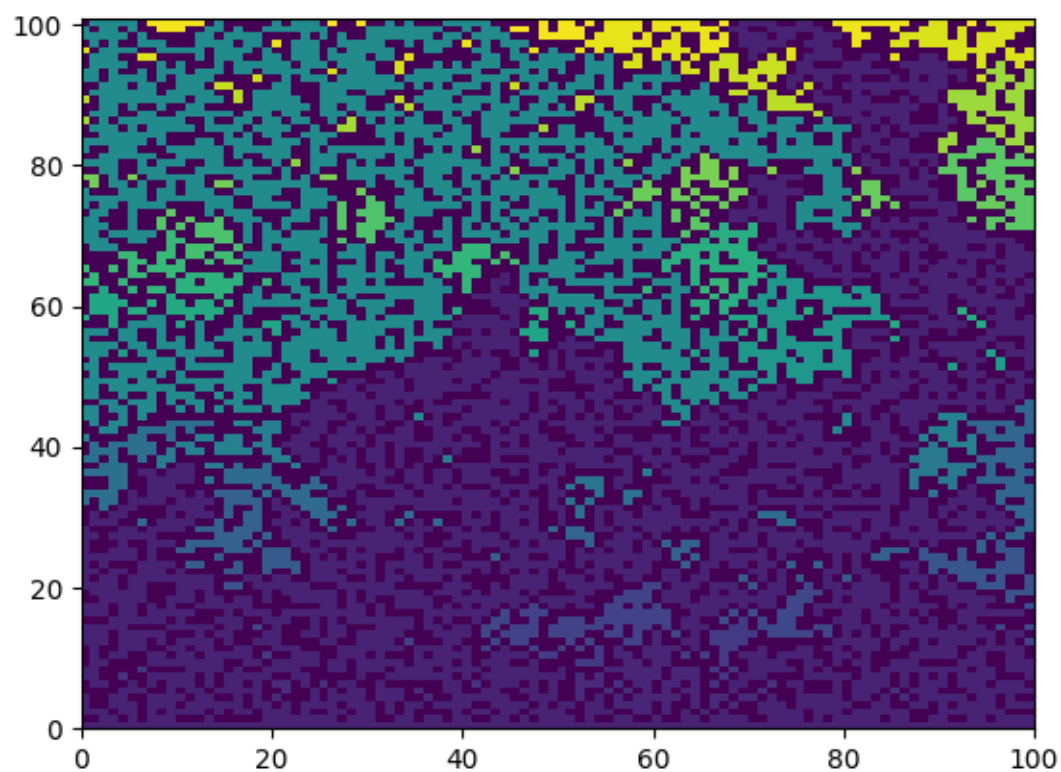
سوال ۴ و ۵:

الگوریتم سوال دقیقاً همان الگوریتم مطرح شده در کلاس می باشد. تشکیل صفحه ای  $L \times L$  با قرار دادن احتمال دلخواه شکل زیر را برای سوال ۴ به دست می آوریم:

برای طول ۱۰۰ و احتمال  $\frac{1}{2}$ :



برای طول ۱۰۰ و احتمال  $\frac{1}{2}$  در سوال ۵:

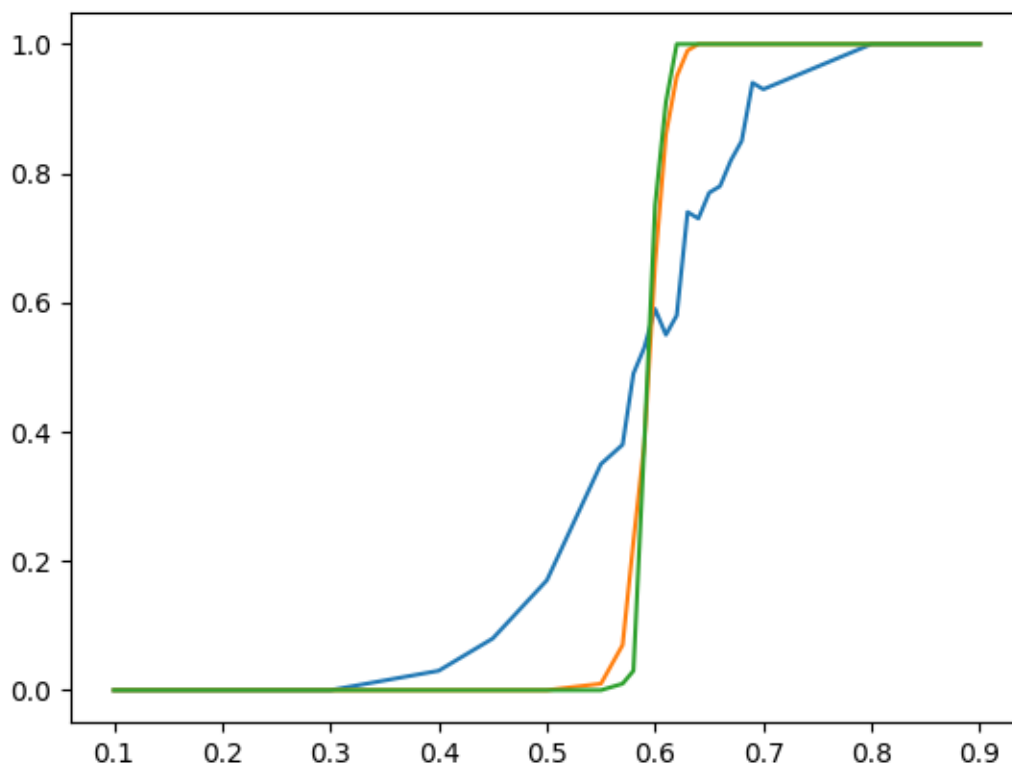


## سوال ۶:

در این سوال با استفاده از الگوریتم رنگ آمیزی و توضیح تابعی برای چک کردن percolation مواجه هستیم. در کنار این دو تابع recolor و color checker را برای تولید نمودار معرفی می‌کنیم. در اینجا نیز همانند سوال ۱ و ۲ نیاز به تعریف چند متغیر Global داریم که بتوانیم در داخل توابع از آنها استفاده کنیم.

نکته قابل توجه در این سوال آن بود که در صورت سوال ذکر شده بود برای طول های ۱۰۰ و ۲۰۰ نیز این کار تکرار شود که برای این مهم، به جای تنظیم چندبار طول، یک آرایه با سه عضو ۱۰، ۱۰۰، ۲۰۰ تعیین کردیم که بتوانیم هر ۳ را در یک نمودار به سادگی مشخص کنیم.

در نمودار به دست آمده خط آبی برای طول ۱۰، خط نارنجی برای طول ۱۰۰ و خط سبز برای طول ۲۰۰ می‌باشد.



سوال ۷:

تفاوت این سوال و سوال قبل تنها در تعیین یک تابع دیگر برای چک کردن رابطه Cluster خواسته شده با Infinity Cluster است. تابع area infinity به همین منظور طراحی شده است. ترکیب رنگ همانند بالا است.

