

بسم الله الرحمن الرحيم

سری چهارم تمرینات درس شبیه سازی فیزیک

حسین محمدی - ۹۶۱۰۱۰۳۵

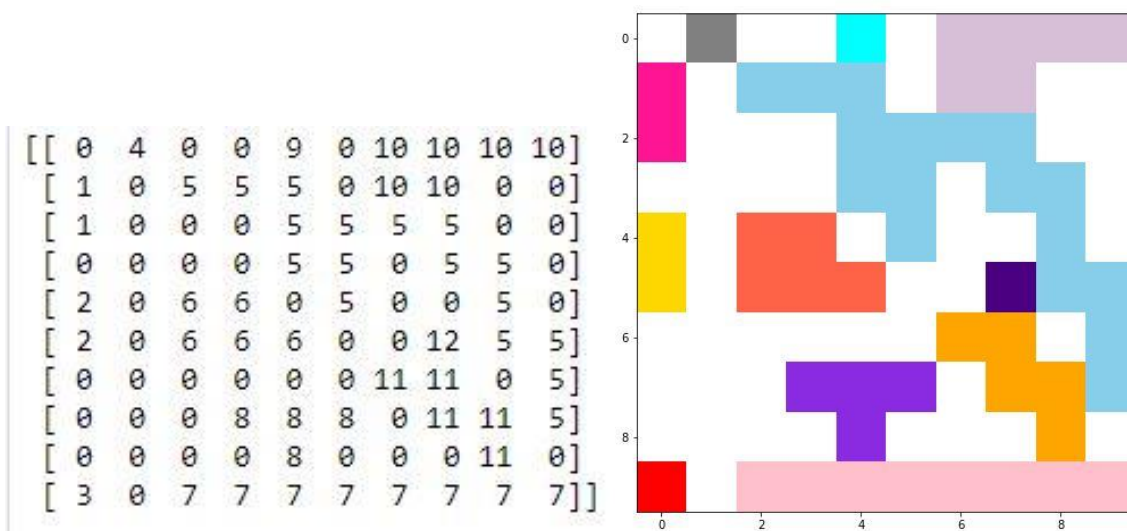
توجه: با کمک متغیرهای اولیه ی کد، گام ها و تعداد خانه ها و... را کنترل کنید، کد برای اجرای کد به کتابخانه های numpy و matplotlib نیاز مند است. تمامی نمودارها با کپشن و لیبل رسم شده اند. برای نمایش شکل در اولین اجرا کد را دو بار ران کنید.

آلگوریتم این تمرین همان آلگوریتم هشن کوپلمن است که نیازی به ندارد، فقط در همین حد بگویم که بنده از روش پیشنهادی آقای سجادی برای پیاده سازی آلگوریتم استفاده کردم، یعنی از تابعی استفاده کردم که برای یافتن هر عدد خانه به کمک حلقه ی While، رئیس خانه را پیدا می کند. شکل کلی این تابع چنین است:

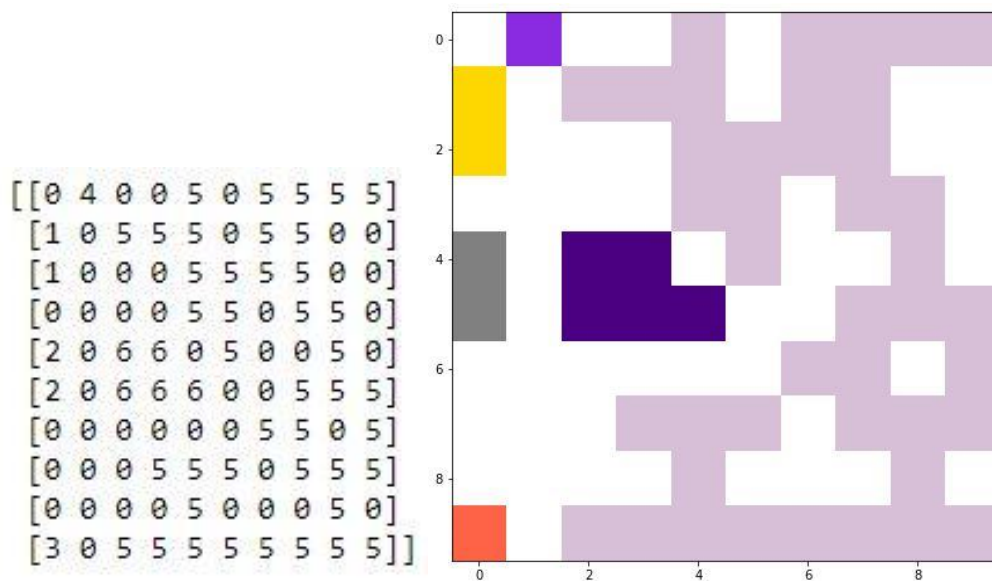
```
def Where_It_Points(L,x):  
    while L[x] != x:  
        x = L[x]  
    return L[x]
```

که در آن L، آرایه ی رنگ های نسبت داده شده و x المنتی است که می خواهیم رئیس اصلی اش را پیدا کنیم. برای متوسط گیری روی مقدار مشخص p و برای پرمایش مقادیر مختلف p نیاز به دو حلقه داریم که گامها را می توانید با متغیر های اولیه تغییر دهید.

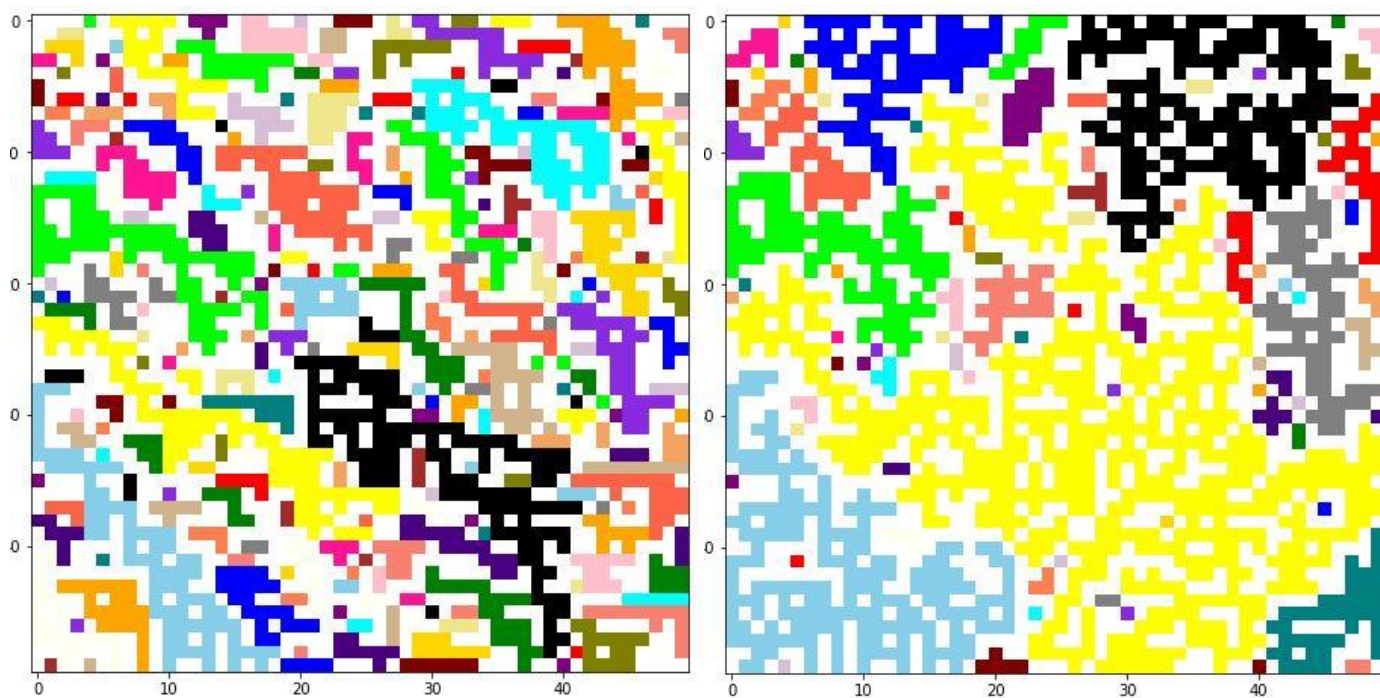
چند نمونه خروجی از کد ببینید: (برای  $L=10$  و  $L=50$ )



(این برای یک آرایه تصادفی با احتمال  $p=0.52$  است، توجه کنید که هنوز رنگ آمیزی صحیح اعمال نشده است و پس از رنگ آمیزی صحیح چنین تصویری بدست می آید.)

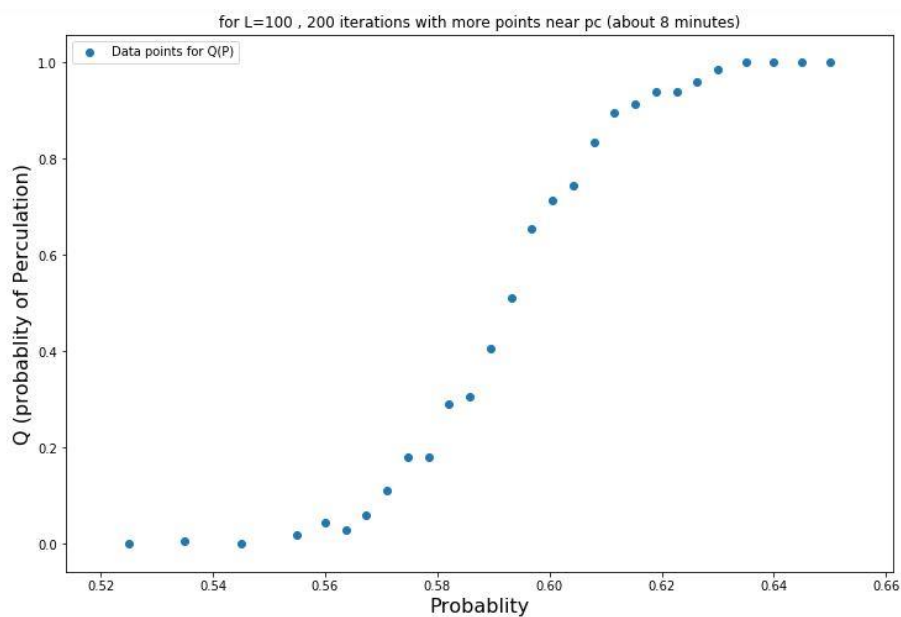
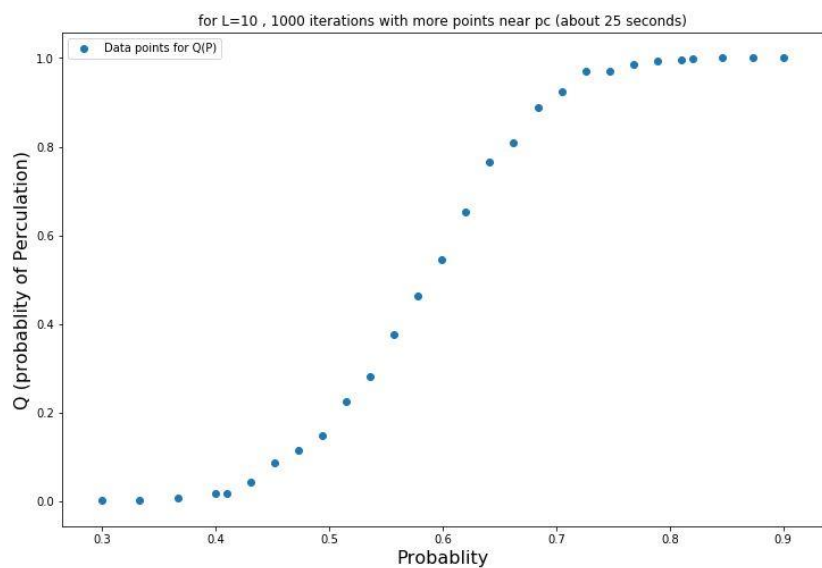


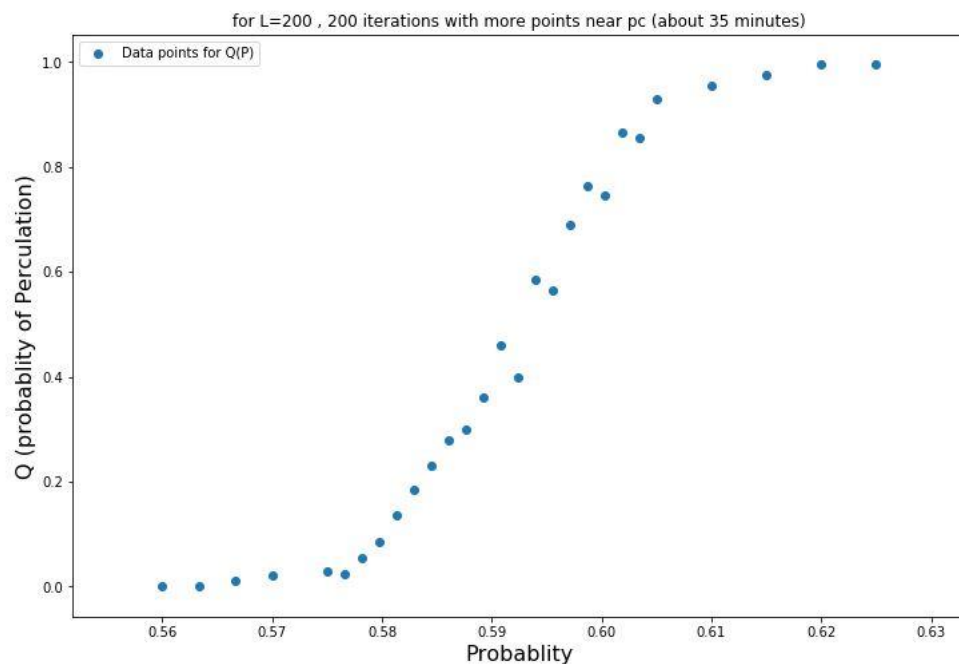
توجه شود که کل حسن الگوریتم هشن کپلمن به این است که آرایه یکبار پرمایش می شود، فلذا برای تولید ماتریس بالا، آرایه یکبار دیگر پرمایش شده تا تک تک درایه ها عدد صحیح بگیرند( این کار را برای چک کردن کد انجام دادم)



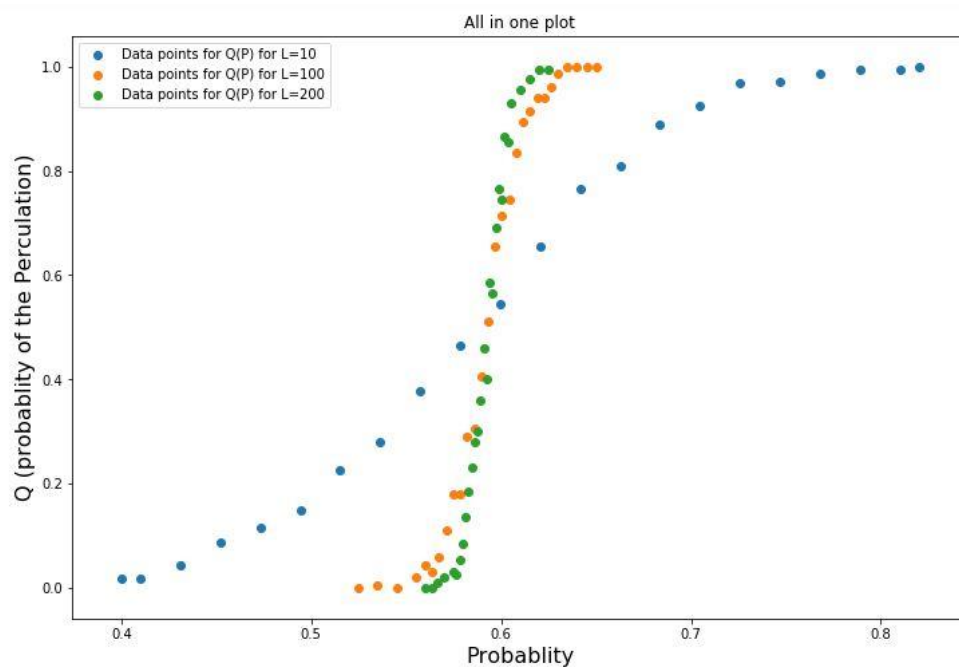
توجه شود که ممکن است برخی خانه ها خالی باشند و در تصویر کناری پر باشند و این به علت این است که به آن ها رنگ سفید تخصیص می یابد(یعنی در کد این خانه ها صفر نیستند، بلکه خالی اند) و علت این باگ هم همانطور که در تمرین های قبلی گفتم، این است که تعداد (خانه های رنگ شده با رنگ متمایز) با تعداد رنگ ها متفاوتند، فلذا برخی رنگ ها تکرار می شوند ( فلذا این را بگذارید به حساب باگهای آگاهانه)

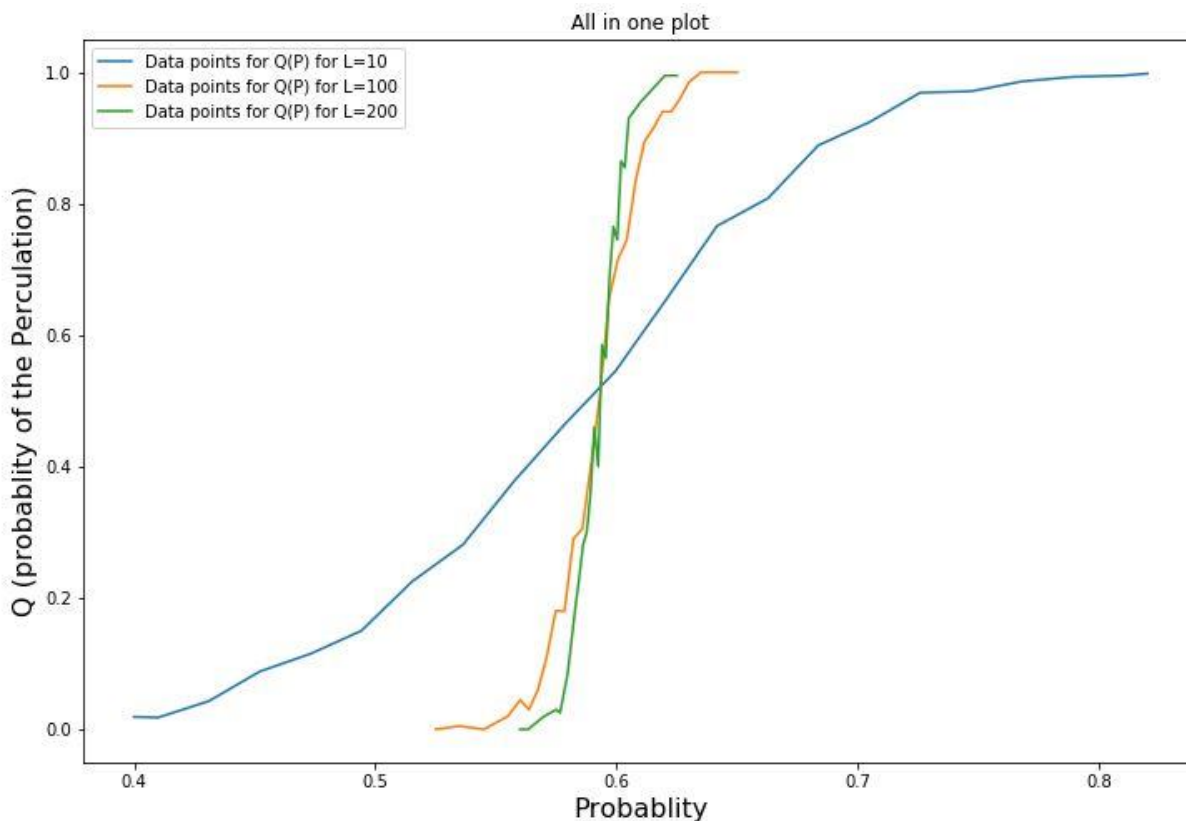
حال برای  $L=10$  و  $L=100$  و  $L=200$  کد را اجرا می کنیم (توجه شود که تعداد گام ها و مقدار اولیه دقیقا همان که در تمرین ذکر شده نیست و علت آن هم دقیقتر شدن نمودار می باشد)





می توان، برای  $L$  های بزرگتر  
تبدیل شدن نمودار  $Q$  بر  
حساب  $P$  را به تابع پله مشاهده  
کرد (این به شخصه برای من  
خیلی جالب بود)  
در آخر هر سه این ها را روی  
یک نمودار رسم می کنیم:





جاهایی که نمودار سبز رنگ و نارنجی رنگ را رسم نکرده ام، تقریباً صفر هستند.

برای  $L$  های کوچکتر، شکل هموارتر و با تندی (شیب) کمتری در نقطه تقعر (تقریباً  $P_c$ ) امتداد می یابد، و برای  $L$  بزرگتر این شیب زیاد می شود و پیش بینی این است که برای حد ترمودینامیکی، تابع به شکل پله می شود، اما در مورد این که چرا برای  $L$  های پائین تر، تراوش برای  $P$  های کمتری رخ می دهد، می توان چنین تصور کرد که شبکه ای بی نهایت داریم و به طور تصادفی یک شبکه به طول ۱۰ در ۱۰ از آن انتخاب می کنیم (شبکه اصلی که دارای طول بی نهایت بود با احتمال  $P$  رسم شده بود) و تراوش برای اعضای گروه بازبهنجارش با طول ۱۰ (که نسبت به ۵۰ و ۱۰۰ بسیار کمتر است، محتمل تر است، پس این که برای  $L$  های کمتر این تراوش برای  $p$  های کمتر رخ دهد و رفته رفته با افزایش طول شبکه این تراوش برای احتمال کمتر رخ ندهد، یک نتیجه منطقی از گروه های بازبهنجارش با طول های متفاوت است.

یک توجیه خوب برای تیز شدن نمودار یا پرشیب تر شدن این است که هر چقدر هم تعداد آنسامبل ها را بیشتر کنیم، چون طول شبکه کم است، نمی توان رفتاری را که برای  $L$  های بی نهایت (حالت ایده آل) استخراج کردیم مشاهده کرد، به زبان شهودی تر، برای ۱۰۰ خانه نمی توان رفتار را مشاهده کرد، اگر  $L$  را زیاد کنیم، این اتفاق بهتر مشاهده می شد.

دیتاپوینت های تمامی آزمایش ها را در فایلی با عنوان Data نوشته ام و در همین فایل زیپ قرار داده ام.