

به نام خدا

محمدرضا دولتی

۹۷۱۱۰۴۱۱

گزارش تمرین عملی آمار و احتمال مهندسی:

تمرین ۱:

برای انجام این تمرین توابع زیر نوشته شده است و سپس از آن ها استفاده شده است:

min_dist: این تابع فاصله نقطه را از هر ضلع حساب می کند و سپس کمترین فاصله را برمی گرداند.

Square: این تابع یک مربع حول نقطه ورودی در نظر می گیرد و سپس یک عدد رندوم را داخل آن مربع در نظر می گیرد و طول ضلع مربع جدید $distance * p$ خواهد بود که **distance** از تابع قبلی محاسبه می شود.

Counter: این تابع سه ورودی دارد. ورودی اول یک ماتریس $n * n$ می باشد که در آن n برابر است با تعداد زیر مربع ها و ورودی های x, y که تعیین می کند این نقطه ورودی متعلق به کدام زیر مربع است. برای این کار کافی است مختصات ضربدر تعداد زیر مربع ها شود.

add_arrow_to_line2D: این تابع برای بخش ب یک سری فلش روی نمودار رسم می کند تا مسیر حرکت مشخص می شود.

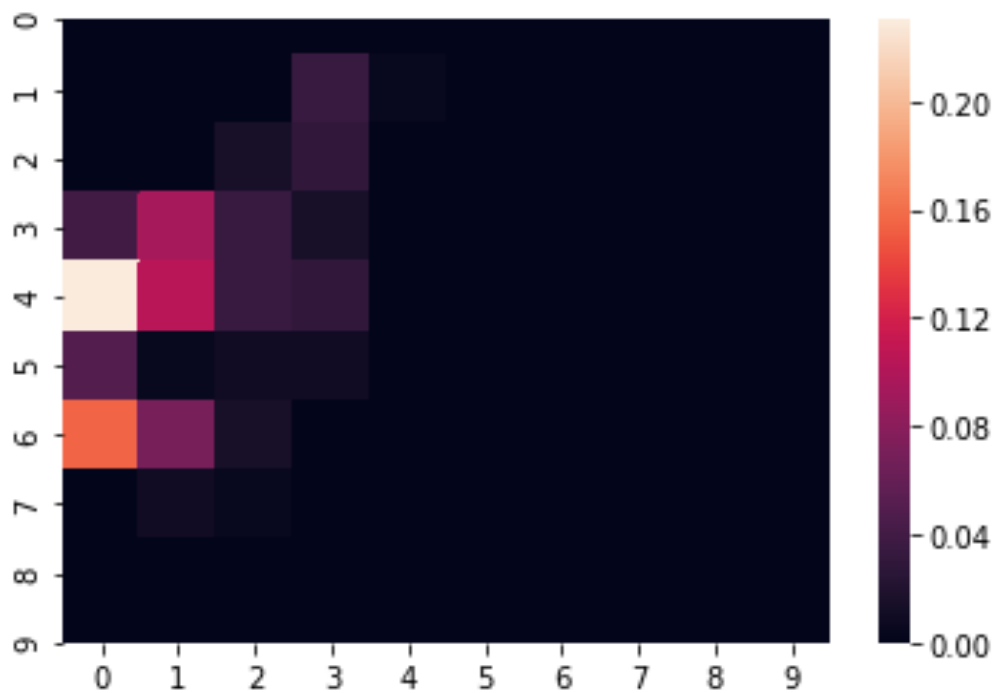
بخش الف:

کد زیر از توابع گفته شده استفاده می کند. روند مسئله به این صورت است که ابتدا یک رندوم شامل مختصات x, y در نظر می گیریم، سپس وارد یک حلقه **for** می شویم. تعیین میکنیم که این نقطه در کجا قرار دارد و سپس با استفاده از تابع **Square** یک مربع به ضلع $dist * p$ دور نقطه ایجاد می کنیم و این کار را تکرار می کنیم و در هر تکرار مختصات زیر را ثبت می کنیم.

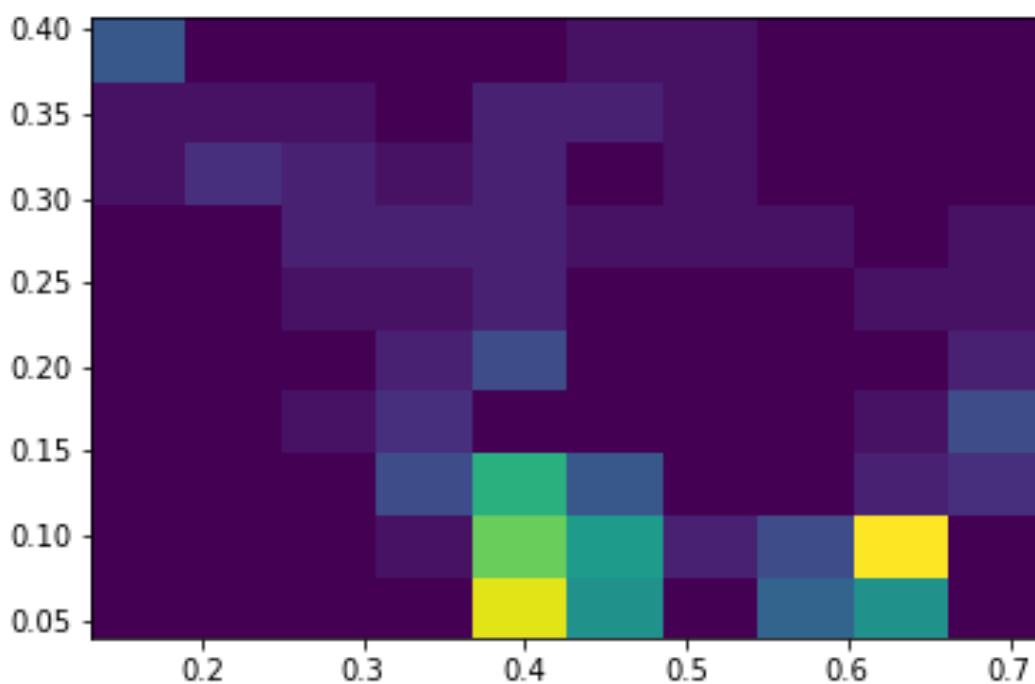
```
p = 1/2;
x = np.random.rand()
y = np.random.rand()
num_bins = 10
max_iters = 200
probs = np.zeros((num_bins, num_bins))

coords_x = []
coords_y = []
dists = []
for i in range(max_iters):
    counter(probs, x, y, num_bins = num_bins)
    coords_x.append(x)
    coords_y.append(y)
    dist = min_dist(x,y)
    dists.append(dist)
    coord, x, y = square(x, y, dist, p = p)
```

برای رسم heatmap از دو روش استفاده کرده‌ایم، یک روش استفاده از کتابخانه seaborn که در واقع احتمال بدست آمده در مرحله قبل (خروجی تابع counter که در واقع هر بار ماتریس prob را بروز می‌کند) را رسم می‌کند:



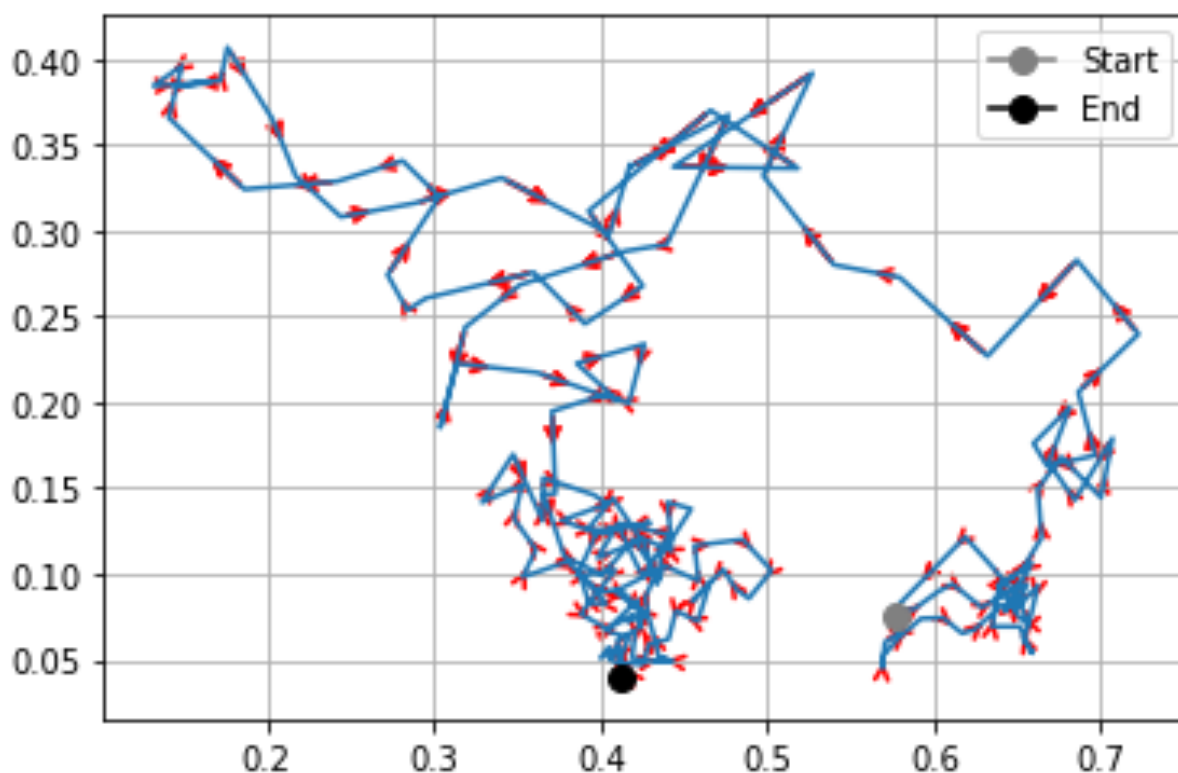
روش دیگر رسم histogram دو بعدی با استفاده از کتابخانه matplotlib است:



بخش ب:

برای این بخش نیز از تابع `add_arrow_to_line2D` استفاده می‌شود، این تابع به صورت کلی به این شکل عمل می‌کند که تابع رسم شده را به عنوان ورودی می‌پذیرد، سپس بین هر دو نقطه در این تابع از شروع تا وسط خط را یک فلش رسم می‌کند.

همچنین برای مشخص بودن نقطه‌ی شروع و پایان مطابق شکل زیر رسم شده است:



در اجرا های مختلف مشاهده شد که همگرایی همواره نزدیک یکی از اضلاع اتفاق می‌افتد.

بخش پ:

برای تعیین زمان اجرا هربار با $100p$ بار تکرار اجرا کردیم و میانگین گرفتیم:

