دانشگاه صنعتی شریف
دانشکده مهندسی برق

عنوان:

گزارش پروژه

اعضای گروه

برنا قاضی زاده - ۴۰۱۱۰۲۲۸۶
محمدپارسا قادراحمدی - ۴۰۱۱۰۲۲۵۳

نام درس

آمار و احتمال مهندسی

نیم‌سال اول ۱۴۰۲-۱۴۰۳

نام استاد درس

محمدحسین یاسائی میبدی

۱ قدم برون نه اگر میل جست و جو داری!

۲ پیش نیازها

۲-۱ پرسش تئوری ۱

اثبات قضیه مارکف:

$$\mathbb{E}[X] = \int_{a}^{\infty} xf_X(x)dx = \int_{-\infty}^{a} xf_X(x)dx + \int_{a}^{\infty} xf_X(x)dx$$

$$\Longrightarrow \mathbb{E}[X] \geqslant \int_{a}^{\infty} xf_X(x)dx$$

$$\int_{a}^{\infty} xf_X(x)dx \geqslant a\int_{a}^{\infty} f_X(x)dx$$

$$\Longrightarrow \mathrm{P}\{X \geqslant a\} \leqslant \frac{\mathbb{E}[X]}{a}$$

۲-۲ پرسش تئوری ۲

۱. X را متغیر تصادفی درصدی از مردم که در یک روز در یک روز در ایستگاه سیرک پیاده میشوند تعریف میکنیم:

$$\mathbb{E}[X \geqslant ۸۵\%] = \frac{۷۵}{۸۵} = \frac{۱۵}{۱۷}$$

۲.

$$\mathrm{P}\{X \geqslant ۶۵\%\} - \mathrm{P}\{X \geqslant ۸۵\%\} = \mathrm{P}\{۶۵\% \leqslant X \leqslant ۸۵\%\}$$

در اینجا حد بازه اهمیتی ندارد چون احتمال یک نقطه از یک بازه پیوسته صفر است.

$$\Longrightarrow \mathrm{P}\{۶۵\% \leqslant X \leqslant ۸۵\%\} \geqslant ۷۵(\frac{۱}{۶۵} - \frac{۱}{۸۵}) = ۱۵(\frac{۱}{۱۳} - \frac{۱}{۱۷})$$

۳. از نامساوی چبیشف برای یافتن کران پایین استفاده میکنیم:

$$\mathrm{P}\{|\frac{\Sigma X_i}{n} - \mu| > \epsilon\} \leqslant \frac{\sigma^۲}{n\epsilon^۲} \Longrightarrow \mathrm{P}\{|\frac{\Sigma X_i}{n} - \mu| \leqslant \epsilon\} \geqslant ۱ - \frac{\sigma^۲}{n\epsilon^۲}$$

$$n \leqslant \frac{\sigma^۲}{۱ - \mathrm{P}\{|\frac{\Sigma X_i}{n} - \mu| \leqslant \epsilon\}\epsilon^۲}$$

۲-۳ پرسش تئوری ۳

۱.

$$Z_n = \sum_{i=1}^{n} X_i \quad , \quad \mathbb{E}[X_i] = \lambda$$

$$\mathbb{E}[Z_n] = \sum_{i=1}^{n} \mathbb{E}[X_i] = n\lambda$$

همچنین از قضیه مارکف داریم:

$$\mathbb{P}[Z_n \geqslant kn\lambda] \leqslant \frac{\mathbb{E}[Z_n]}{kn\lambda} = \frac{1}{k}$$

۲. متغیر تصادفی Y را به صورت زیر تعریف می کنیم:

$$Y \triangleq \frac{Z_n - n\mu}{\sqrt{n}\sigma} \qquad \underset{\text{CLT}}{\Longrightarrow} \qquad Y \sim \mathcal{N}(0, 1)$$

$$X_i \sim Poisson(\lambda) \Rightarrow \mu = \lambda \quad , \quad \sigma = \sqrt{\lambda}$$

$$\begin{aligned}
\mathbb{P}[Z_n \geqslant kn\lambda] &= \mathbb{P}[\frac{Z_n - n\lambda}{\sqrt{n\lambda}} \geqslant \frac{kn\lambda - n\lambda}{\sqrt{n\lambda}}] \\
&= \mathbb{P}[Y \geqslant \sqrt{n\lambda}(k-1)] \\
&= \frac{1}{\sqrt{2\pi}} \int_{\sqrt{n\lambda}(k-1)}^{\infty} e^{-\frac{x^2}{2}} dx \\
&= \frac{1}{2\sqrt{2}} \frac{2}{\sqrt{\pi}} [\int_0^{\infty} e^{-\frac{x^2}{2}} dx - \int_0^{\sqrt{n\lambda}(k-1)} e^{-\frac{x^2}{2}} dx \\
&= \frac{1}{2}[erf(\infty) - erf(\sqrt{\frac{n\lambda}{2}}(k-1))] \\
&= \frac{1}{2}[1 - erf(\frac{\sqrt{10}}{4})] \approx 0/132
\end{aligned}$$

۲-۴ پرسش تئوری ۴

$$Z_n = \sum_{i=1}^{n} X_i \Longrightarrow \psi_{Zn}(s) = \psi_{\Sigma} X_i(s) = \ln \mathbb{E}[e^{s(X_1+X_2+...+X_n)}]$$

چون X_i ها هم توزیع و مستقل از هم اند:

$$\psi_{Zn}(s) = \ln \mathbb{E}[e^{sX_1} e^{sX_2} ... e^{sX_n}]$$

$$\psi_{Zn}(s) = \ln \mathbb{E}[e^{sX_1}] \ln \mathbb{E}[e^{sX_2}] ... \ln \mathbb{E}[e^{sX_n}] = \ln \mathbb{E}[e^{sX_1}]^n = n \ln \mathbb{E}[e^{sX_1}] = n\psi_X(s)$$

۲-۵ پرسش تئوری ۵

$$\mathrm{P}\{Z_n \geqslant \beta n\} = \mathrm{P}\{e^{sZ_n} \geqslant e^{s\beta n}\}$$

$$\mathrm{P}\{e^{sZ_n} \geqslant e^{s\beta n}\} \leqslant \frac{\mathbb{E}[e^{sZ_n}]}{e^{s\beta n}}$$

$$\psi_{Zn}(s) = \ln \mathbb{E}[e^{sZ_n}] \Longrightarrow e^{\psi_{Zn}(s)} = \mathbb{E}[e^{sZ_n}]$$

$$\mathrm{P}\{e^{sZ_n} \geqslant e^{s\beta n}\} \leqslant e^{\psi_{Zn}(s)-s\beta n} = e^{n(\psi_X(s)-\beta s)}$$

میدانیم نامساوی بالا به ازای هر مقدار مثبتی از s برقرار است و زمانی مینیمم میشود که $\beta s - \psi_X(s)$ بیشینه شود:

$$r = \sup_{s\geqslant ۰}\{\beta s - \psi_X(s)\}$$

$$\mathrm{P}\{Z_n \geqslant \beta n\} = \mathrm{P}\{e^{sZ_n} \geqslant e^{s\beta n}\} \leqslant e^{-rn}$$

۲-۶ پرسش تئوری ۶

۱. همانطور که سر کلاس اثبات شد، می دانیم تابع مولد گشتاور توزیع گاوسی از رابطه زیر پیروی می کند:

$$\Phi_X(s) = e^{\mu s + \frac{\sigma^2 s^2}{2}}$$

با لگاریتم گرفتن از تابع بالا به نتیجه زیر می رسیم:

$$\psi_X(s) = \mu s + \frac{\sigma^2 s^2}{2}$$

۲. با استفاده از نتیجه بخش قبل داریم:

$$\psi_X(s) = \frac{Var(X)s^2}{2}$$

و نیز با توجه به این نکته که

$$Var(X) = \mathbb{E}[X^2] - \mathbb{E}^2[X] = \mathbb{E}[X^2]$$

و استفاده از فرض صورت سوال

$$|X| < M \Rightarrow \qquad Var(X) < M^2$$

به نتیجه صورت سوال خواهیم رسید:

$$\psi_X(s) < \frac{1}{2}M^2\sigma^2$$

۳. در این بخش از یافته های بخش های قبل استفاده می کنیم. از نتیجه سوال قبل می دانیم:

$$\mathbb{P}[Z_n \geqslant n\beta] \leqslant e^{-n\sup_{s\geqslant 0}\{s\beta - \psi_X(s)\}} = e^{-n\sup_{s\geqslant 0}\{s\beta - s\mu - \frac{\sigma^2 s^2}{2}\}}$$

که تابعی که از آن سوپریمم گرفته می شود، یک سهمی محدب است. در نتیجه ماکسیمم در

$-\frac{b}{2a}$ رخ می دهد که در آن b و a پارامتر های سهمی ما هستند. در نتیجه:

$$s^* = \arg\max_s (s(\beta - \mu) - \frac{\sigma^2 s^2}{2}) = \frac{\beta - \mu}{\sigma^2}$$

در نتیجه با جایگذاری :

$$\mathbb{P}[Z_n \geqslant n\beta] \leqslant e^{-n\frac{(\beta-\mu)^2}{2\sigma^2}}$$

از طرفی نیز:

$$\sigma^2 = \mathbb{E}[X^2] - \mathbb{E}^2[X] \leqslant M^2 - \mu^2$$

$$\Rightarrow -\frac{1}{\sigma^2} \leqslant -\frac{1}{M^2 - \mu^2}$$

$$\Rightarrow e^{-n\frac{(\beta-\mu)^2}{2\sigma^2}} \leqslant e^{-n\frac{(\beta-\mu)^2}{2(M^2-\mu^2)}}$$

و با قرار دادن $\mu = 0$ به نتیجه زیر می رسیم:

$$\mathbb{P}[Z_n \geqslant n\beta] \leqslant e^{-n\frac{\beta^2}{2M^2}}$$

۳ که دراز است رَهِ مقصد و من نوسفرم!

۳-۱ پرسش تئوری ۷

۱. اگر Z_n شماره ایستگاه پس از n مرحله باشد، داریم:

$$Z_n = ۱ + \sum_{i=۱}^{n} X_i \Longrightarrow \mathrm{P}\{Z_n = k\} = \binom{n}{k-۱} P^{k-۱}(۱-p)^{n-p+۱}$$

در معادله‌ی بالا، عدد یک شماره‌ی ایستگاه اول است.
در واقع Z_n را میتوان یک متغیر $binomial$ دانست و به آن مدل کرد.

۲. در واقع هرکدام از X_i ها یک متغیر تصادفی برنولی با پارامتر p هستند.

$$\mathbb{E}[X_i] = p, Var(X_i) = p(۱-p)$$

۳. همانطور که گفتیم، Zn را میتوان یک متغیر $binomial$ و یک مقدار ثابت مدل کرد:

$$\mathbb{E}[Z_n] = \sum_{i=۱}^{n} X_i + \mathbb{E}[Z_۰] = ۱ + np$$

$$Var(Z_n) = Var(\sum_{i=۱}^{n} X_i + Z_۰) = Var(\sum_{i=۱}^{n} X_i) = np(۱-p)$$

۴. $\mathbb{E}[Z_n]$ ها در واقع نشان می دهد به طور میانگین چنگیز در چه ایستگاهی پیاده میشود.

۲-۳ پرسش تئوری ۸

مطلوب سوال این است که مجموع امتیاز هایش طی حداقل ۸۰ تاس انداختن بیش از ۳۰۰ می شود.
در نتیجه طی ۷۹ پرتاب اول، کمتر از ۳۰۱ است:

$$\sum_{i=1}^{79} X_i < 301$$

$$\Rightarrow \mathbb{E}(X_i) = \frac{1}{10}\sum_{i=1}^{10} i = 5.5$$

$$\Rightarrow Var(X_i) = \frac{1}{10}\sum_{i=1}^{10} i^2 - (5.5)^2 = 38.5 - 30.25 = 8.25$$

$$\begin{aligned}
\mathbb{P}(\sum_{i=1}^{79} X_i < 301) &= \mathbb{P}(\sum_{i=1}^{79} X_i - 79 \times 5.5 < -133.5) \\
&= \mathbb{P}(\frac{\sum_{i=1}^{79} X_i - 79 \times 5.5}{\sqrt{79(8.25)}} < \frac{-133.5}{\sqrt{79(8.25)}}) \\
&= \Phi(\frac{-133.5}{\sqrt{79(8.25)}})
\end{aligned}$$

۳-۳ پرسش تئوری ۹

۱.

$$P_{Zn}(z) = \frac{۱}{۲}P_{Zn-۱}(z-۱) + \frac{۱}{۲}P_{Zn-۱}(z+۱)$$

$P_{Zn-۱}(z-۱) \longrightarrow$ یک ایستگاه جلو آمده

$P_{Zn-۱}(z+۱) \longrightarrow$ یک ایستگاه عقب رفته

۲. باید به صورت بازگشتی جملات را بنویسیم:

$$P_{Z_n}[z] = \frac{۱}{۲}[P_{Z_{n-۱}}[z-۱] + P_{Z_{n-۱}}[z+۱]]$$

$$P_{Z_n}[z] = \frac{۱}{۲^۲}[P_{Z_{n-۲}}[z-۲] + P_{Z_{n-۲}}[z+۲] + ۲P_{Z_{n-۲}}[z]]$$

$$P_{Z_n}[z] = \frac{۱}{۲^۳}[P_{Z_{n-۳}}[z-۳] + P_{Z_{n-۳}}[z+۳] + ۳P_{Z_{n-۳}}[z+۱] + ۳P_{Z_{n-۳}}[z-۱]]$$

$$\ldots$$

$$P_{Z_n}[z] = \frac{۱}{۲}\sum_{i=۰}^{n}\binom{n}{i}P_{Z_۰}[z-n+۲i]$$

۳. اگر فرض کنیم تعداد حرکات به سمت راست را r و تعداد حرکات به سمت چپ را l بگذاریم، داریم:

$$\begin{cases} r+l=n, \\ r-l=z \qquad \mathrm{z} > ۰ \ . \end{cases} \Longrightarrow r = \frac{n+z}{۲}, l = \frac{n-z}{۲} \Longrightarrow$$

عملا یک فرایند *binomial* داریم. یک مجموعه ای از حرکات انجام شده و ما میخواهیم r تا از یک چیز و l تای آن چیز دیگری باشند:

$$\mathrm{P}\{Z_n = z\} = \binom{n}{\frac{n+z}{۲}}p^{\frac{n+z}{۲}}q^{\frac{n-z}{۲}} = \frac{n!}{(\frac{n+z}{۲})!(\frac{n-z}{۲})}p^{\frac{n+z}{۲}}q^{\frac{n-z}{۲}}$$

۳-۴ پرسش تئوری ۱۰

ابتدا یک معادله بازگشتی برای احتمال حضور در ایستگاه n ام مینویسیم:

$$P_n = \frac{۱}{۲}P_{n-۱} + \frac{۱}{۲}P_{n+۱}$$

سپس از بخش دوم برای حل این معادله کمک میگیریم:

$$X^۲ - ۲X + ۱ = ۰ \Longrightarrow (X-۱)^۲ = ۰ \Longrightarrow X = ۱$$

حال باید شرایط مرزی برای معادلهی $P_n = A + B_n$ اعمال کنیم.
فرض میکنیم در باغ وحش متوقف شویم:

$$\begin{cases} P_۲ = ۱ = A + ۲B, \\ P_{-}۱ = ۰ = A - B. \end{cases} \Longrightarrow B = \frac{۱}{۳}, A = \frac{۱}{۳} \Longrightarrow P_n = \frac{۱}{۳} + \frac{n}{۳}$$

میدانیم ما از ایستگاه صفر شروع میکنیم، پس $n = ۰$.

$$P_۰ = \frac{۱}{۳}$$

حالتی که در شهربازی متوقف میشیم مکمل حالت فوق است که در بالا حساب کردیم:

$$P_۰ = ۱ - \frac{۱}{۳} = \frac{۲}{۳}$$

۵-۳ پرسش تئوری ۱۱

۱. مانند سوال قبل ابتدا معادله بازگشتی مینویسیم:

$$P_n = pP_{n+1} + qP_{n-1} \Longrightarrow P_n = \lambda^n \Longrightarrow p\lambda^2 - \lambda + q = 0 \Longrightarrow \lambda = \frac{1 \pm \sqrt{1-4pq}}{2p}$$

حال باید روی p حالت بندی کنیم:

اگر $p = \frac{1}{2}$ باشد، داریم:

$$\lambda^2 - 2\lambda + 1 = 0 \Longrightarrow (\lambda - 1)^2 = 0$$

$$P_n = A + B_n \Longrightarrow \begin{cases} P_l = 1 = A + B_l, \\ P_0 = 0 = A. \end{cases} \rightarrow B = \frac{1}{l} \Longrightarrow P_n = \frac{n}{l}$$

حال فرض میکنیم $p \neq \frac{1}{2}$ باشد:

$$\lambda = \frac{1 \pm \sqrt{1 - 4p(1-p)}}{2p} = \frac{1 \pm (2p-1)}{2p} \Longrightarrow \lambda_1 = 1, \lambda_2 = \frac{q}{p}$$

$$P_n = A + B(\frac{q}{p})^n \Longrightarrow \begin{cases} P_l = 1 = B(\frac{q}{p})^l, \\ P_0 = 0 = A + B. \end{cases} \rightarrow A = -B \Longrightarrow A(p^l - q^l) = p^l \Longrightarrow A = \frac{p^l}{p^l - q^l}$$

$$\rightarrow P_n = \frac{p^n - q^n}{p^l - q^l}$$

۲. برای حل این بخش از $conditional\ expected\ value$ استفاده میکنیم:

$$\mathbb{E}[t_n] = p(1 + \mathbb{E}[t_{n+1}]) + q(1 + \mathbb{E}[t_{n-1}]) = (p+q) + p\,\mathbb{E}[t_{n+1}] + q\,\mathbb{E}[t_{n-1}]$$

$$\Longrightarrow \mathbb{E}[t_n] = 1 + p\,\mathbb{E}[t_{n+1}] + q\,\mathbb{E}[t_{n-1}]$$

چون معادله‌ی فوق ناهمگن است باید از روش ناهمگن حل شود، درواقع باید یکبار عدد ۱ را درنظر نگیریم و پاسخ همگن را محاسبه کنیم و سپس جواب مخصوص معادله را بیابیم.

پاسخ معادله‌ی همگن را که عملا از بخش قبل داریم. اگر $p = \frac{۱}{۲}$ باشد، ریشه مضاعف است:

$$\mathbb{E}[t_n] = A + B_n$$

حال باید معادله‌ی ناهمگن را حل کنیم:

$$\mathbb{E}[t_n] = ۱ + p\,\mathbb{E}[t_{n+۱}] + q\,\mathbb{E}[t_{n-۱}]$$

اگر عدد ثابت بگذاریم جواب نداریم، پس مثلا αn میگذاریم:

$$\alpha n = ۱ + p\alpha(n+۱) + q\alpha(n-۱) \longrightarrow ۱ + \alpha p - \alpha q = ۰ \Longrightarrow (p = q = \frac{۱}{۲})$$

سپس $\alpha n^۲$ میگذاریم:

$$\alpha n^۲ = ۱ + p\alpha(n+۱)^۲ + q\alpha(n-۱)^۲ \longrightarrow (۱+\alpha)۲p\alpha - ۲q\alpha = ۰ \Longrightarrow (\alpha = ۱)$$

$$\mathbb{E}[t_n] = A + B_n - n^۲ \Longrightarrow \begin{cases} \mathbb{E}[t_۰] = ۰ = A, \\ \mathbb{E}[t_l] = ۰ = Bl - l^۲. \end{cases} \longrightarrow B = l \Longrightarrow \mathbb{E}[t_n] = nl - n^۲$$

حال فرض میکنیم $p \neq \frac{۱}{۲}$ باشد، آنگاه:

$$\lambda = \frac{۱ \pm \sqrt{(۲p-۱)^۲}}{۲} \longrightarrow \lambda_۱ = p, \lambda_۲ = q$$

اول معادله همگن را حل میکنیم:

$$\mathbb{E}[t_n] = Ap^n + Bq^n$$

حال به بخش ناهمگن می پردازیم، اگر عدد ثابت بگذاریم جواب نداریم پس $\mathbb{E}[t_n] = \alpha n$:

$$\alpha = ۱ + p(\alpha) + g(\alpha) \Longrightarrow ۰ = ۱$$

$$\alpha n = ۱ + p\alpha(n+۱) + g\alpha(n-۱) \longrightarrow p\alpha - q\alpha = ۱ \Longrightarrow \alpha = \frac{۱}{p-q}$$

$$\mathbb{E}[t_n] = Ap^n + Bq^n + \frac{n}{p-q} \Longrightarrow \begin{cases} \mathbb{E}[t_\circ] = \circ = A + B, \\ \mathbb{E}[t_l] = \circ = Ap^l + Bq^l + \frac{l}{p-q}. \end{cases}$$

$$\longrightarrow A(p^l - q^l) = \frac{l}{q-p} \Longrightarrow A = \frac{l}{(q-p)(p^l - q^l)}$$

$$\Longrightarrow \mathbb{E}[t_n] = \frac{l}{(p-q)(p^l - q^l)}(q^n - p^n) + \frac{n}{p-q}$$

حال میخواهیم $\mathbb{E}[t_n]$ بیشینه شود($\frac{\partial}{\partial t}\mathbb{E}[t_n] = \circ$).

اگر $p = \frac{۱}{۲}$ باشد:

$$\mathbb{E}[t_n] = nl - n^{۲} \longrightarrow \frac{\partial}{\partial t}\mathbb{E}[t_n] = l - ۲n \Longrightarrow n = \frac{l}{۲}$$

اگر بخواهیم دقت به خرج بدیم باید بررسی کرد l زوج است یا نه و از طرفی چک کرد $[\frac{l}{۲}]$ و $[\frac{l}{۲}] + ۱$ اگر زوج l نباشد کدام $\mathbb{E}[t_n]$ را ماکسیمم میکند.

$$\Longrightarrow \frac{\partial}{\partial t}\mathbb{E}[t_n] = \frac{nl}{(p-q)(p^l - q^l)}(q^{n-۱} - p^{n-۱}) + \frac{۱}{p-q} = \circ$$

۳-۶ پرسش تئوری ۱۲

.۱

$$P_n = pP_{n+1} + (1-p)P_{n-1} \longrightarrow P_n = \lambda^2 - \lambda + (1-p)\lambda = 0$$

باز هم دو حالت داریم، اگر $p = \frac{1}{2}$:

$$\lambda^2 - 2\lambda + 1 = 0 \longrightarrow (\lambda - 1)^2 = 0 \Longrightarrow \lambda = 1$$

$$P_n = A + B_n \longrightarrow P_0 = 1, A = 1, P_\infty = 1 + B_\infty = 0 \Longrightarrow B = \frac{-1}{\infty}$$

در واقع عملا B، صفر می شود.

$P_n = 1 \Longleftarrow$ در واقع احتمال خارج شدن از مترو فارغ از ایستگاهی که وارد میشود یک است.

$p \neq \frac{1}{2}$ باشد:

$$\lambda = \frac{1 \pm (2p-1)}{2p} \Longrightarrow \lambda_1 = 1, \lambda_L = \frac{q}{p}$$

$$P_n = A + B(\frac{q}{p})^n \longrightarrow P_0 = A + B = 1 \longrightarrow B = 1 - A \Longrightarrow P_n = A + (1-A)(\frac{q}{p})^n$$

$$P_n = A(1 - (\frac{q}{p})^n) + (\frac{q}{p})^n$$

حال مسئله را به دو قسمت تقسیم میکنیم:

اگر $p > \frac{1}{2}$ باشد:

$$q < \frac{1}{2} \longrightarrow \frac{q}{p} < 1 \Longrightarrow \lim_{n\to\infty} (\frac{q}{p})^n = 0$$

$$P_\infty = 0 = A(1-0) + 0 \longrightarrow B = 1 \Longrightarrow (\frac{q}{p})^n = P_n$$

اگر $p < \frac{1}{2}$ باشد:

$$q > \frac{1}{2} \longrightarrow \frac{q}{p} > 1 \Longrightarrow \lim_{n\to\infty} (\frac{q}{p})^n = \infty$$

$$P_\infty = 0 = (\frac{q}{p})^n(1-A) + A \longrightarrow A = 1 \Longrightarrow P_n = 1$$

بله اگر با احتمال بیشتری به سمت جلو حرکت کنیم، ممکن است هیچ وقت خارج نشویم.

.۲

$$\mathbb{E}[t_n] = p(1 + \mathbb{E}[t_{n+1}]) + q(1 + \mathbb{E}[t_{n-1}]) = (p+q) + p\,\mathbb{E}[t_n + 1] + q\,\mathbb{E}[t_n - 1]$$

$$\mathbb{E}[t_n] = ۱ + p\,\mathbb{E}[t_n + ۱] + q\,\mathbb{E}[t_n - ۱]$$

مثل بخش قبل باید از روشی برای حل معادله‌ی غیر همگن استفاده کنیم.
از پاسخ های بخش قبل استفاده میکنیم، اگر $p = \frac{۱}{۲}$ باشد:

$$\mathbb{E}[t_n] = A + B_n - n^۲ \longrightarrow \mathbb{E}[t_۰] = A + B(۰) - ۰ \Longrightarrow A = ۰, \mathbb{E}[t_\infty] = \infty = B(\infty) - (\infty)^۲$$

نمی توان نظر قطعی درمورد B داد.
حال اگر $p \neq \frac{۱}{۲}$ باشد:

$$\mathbb{E}[t_n] = A + B(\frac{q}{p})^n + \frac{n}{p-q} \Longrightarrow p > q \longrightarrow \frac{q}{p} < ۱ \Longrightarrow \mathbb{E}[t_۰] = ۰ = A + B, \mathbb{E}[t_\infty] = \infty$$

$$\mathbb{E}[t_n] = A + B(\frac{q}{p})^n + \frac{n}{p-q} = A + B(۰) + \frac{\infty}{p-q}$$

باز هم نمیتوان درباره A ، B نظر قطعی داد.

۳-۷ پرسش تئوری ۱۳

۱. Y_n را متغیر تصادفی تعریف میکنیم که در یک حرکت n گام برداشته شود.
این حرکت برابر با جمع n متغیر تصادفی تک گام است یعنی:

$$Y_n = X_1 + X_2 + X_3 + ... + X_n$$

۲. زیرا احتمال اینکه زمانی بهع ایستگاه یک برسد برابر است با احتمال اینکه زمانی به ایستگاه دو و چپ رود یا زمانی که به ایستگاه صفر برسد و راست رود.

$$P_1 = pP_0 + qP_2$$

ولی چون در اولین گام به ایستگاه صفرم میرویم پس $P_0 = 1$:

$$P_1 = p + qP_2$$

۳. اگر یک گام جلو برویم آنگاه از خانه ۱ ، انگار مسئله مجدداً تکرار میشود. پس با احتمال P_1 به خانه ۱ رسیده و پس از آن مجدد با احتمال P_1 به خانه ۲ میرسیم: phi

$$P_2 = P_1^2$$

$$P_1 = p + qP_1^2$$

$$qP_1^2 - P_1 + p = 0 \longrightarrow P_1 = \frac{1 \pm \sqrt{1 - 4qp}}{2q} = \frac{1 \pm (1 - 2p)}{2q} \Longrightarrow P_1 = \begin{cases} \frac{p}{q}, \\ 1. \end{cases}$$

۴. برای $k > 0$:

$$P_k = P_1^k, P_0 = 1 \Longrightarrow P_{-k} = P_{-1}^k$$

بررسی حالات مختلف برای p:

اگر $p \geqslant \frac{۱}{۲}$:

$$P_k = \begin{cases} ۱ & \text{k} \geqslant ۰. \\ (\frac{q}{p})^{-k} & \text{k} < ۰ \end{cases}$$

اگر $p > \frac{۱}{۲}$:

$$P_k = \begin{cases} ۱ & \text{k} \leqslant ۰. \\ (\frac{p}{q})^{k} & \text{k} > ۰ \end{cases}$$

۵. اگر $p = \frac{۱}{۲}$ باشد، با توجه به توضیحات بالا احتمال رسیدن به همه ایستگاه ها برابر با ۱ است.

۶.

$$p = ۰/۳ \longrightarrow p < \frac{۱}{۲} \Longrightarrow P_۵ = (\frac{۰/۳}{۰/۷})^۵ = (\frac{۳}{۷})^۵ \simeq ۰/۰۱۴$$

۳-۸ پرسش تئوری ۱۴

۱. مدت زمانی که طول میکشد از ایستگاه $k-1$ به ایستگاه k برویم، برابر با مدت زمانی که طول میکشد از ایستگاه صفرم به ایستگاه یکم برویم.در نتیجه:

$$\mathbb{E}[T_k] = \mathbb{E}[T_{k-1}] + \mathbb{E}[T_1]$$

$$\mathbb{E}[T_{k-1}] = \mathbb{E}[T_{k-2}] + \mathbb{E}[T_1]$$

$$\cdots$$

$$\mathbb{E}[T_2] = \mathbb{E}[T_1] + \mathbb{E}[T_1]$$

اگر عبارات فوق را باهم جمع کنیم:

$$\mathbb{E}[T_k] = k\,\mathbb{E}[T_1]$$

۲.

$$\mathbb{E}[T_1] = p(1+\mathbb{E}[T_0]) + q(1+\mathbb{E}[T_2]) = p(1+0) + q(1+\mathbb{E}[T_2]) = 1 + q\,\mathbb{E}[T_2]$$

$$\mathbb{E}[T_2] = 2\,\mathbb{E}[T_1] = \frac{\mathbb{E}[T_1]-1}{q}$$

$$\Longrightarrow \mathbb{E}[T_1] = \frac{1}{2p-1} = \frac{1}{1-2q}$$

۳.

$$\mathbb{E}[T_k] = \frac{k}{2p-1}$$

۴.

$$\mathbb{E}[T_{50}] = \frac{50}{0/1} = 500$$

۳-۹ پرسش تئوری ۱۵

$$Z_n = Z_0 + \sum_{i=1}^{n} X_i$$

$$\mathrm{P}\{Z_{10} > z\} = \mathrm{P}\{\sum_{i=1}^{n} X_i > z - Z_0\} = \mathrm{P}\{\sum_{i=1}^{n} X_i > 5\} = \mathrm{P}\{\frac{\sum X_i}{\sqrt{10}} > \frac{5}{\sqrt{10}}\}$$

$$\mathrm{P}\{Z_{10} > z\} = 1 - \phi(\frac{5}{\sqrt{10}})$$

۳-۱۰ پرسش تئوری ۱۶

۳-۱۱ پرسش تئوری ۱۷

۳-۱۲ پرسش تئوری ۱۸

۳-۱۳ پرسش تئوری ۱۹

.۱

$$\begin{aligned}
&\mathbb{E}[Z_n] = \sum_{i=۱}^{n} \mathbb{E}[X_i] = \sum_{i=۱}^{n} \begin{bmatrix} r\mathbb{E}[\cos\theta_i] \\ r\mathbb{E}[\sin\theta_i] \end{bmatrix} \\
&\mathbb{E}[\cos\theta] = \int_{۰}^{۲\pi} \cos\theta \frac{d\theta}{۲\pi} = ۰ \\
&\mathbb{E}[Z_n] = ۰
\end{aligned}$$

$$Z_n = r \begin{bmatrix} \sum_{i=۱}^{n} \cos\theta_i \\ \sum_{i=۱}^{n} \sin\theta_i \end{bmatrix}$$

.۲

$$\begin{aligned}
||Z_n||^۲ &= r^۲((\sum_{i=۱}^{n} \cos\theta_i)^۲ + (\sum_{i=۱}^{n} \sin\theta_i)^۲) \\
&= r^۲(n + ۲\sum_{i=۱}^{n-۱} \sum_{j=i+۱}^{n} (\cos\theta_i \cos\theta_j + \sin\theta_i \sin\theta_j))
\end{aligned}$$

$$\forall i \neq j: \qquad \mathbb{E}[\cos\theta_i \cos\theta_j] = \mathbb{E}[\cos\theta_i]\mathbb{E}[\cos\theta_j] = ۰ = \mathbb{E}[\sin\theta_i \sin\theta_j]$$

$$\Rightarrow \mathbb{E}(||Z_n||^۲) = nr^۲$$

.۳

$$\mathbb{E}[\cos^۲\theta_i] = \frac{۱}{۲\pi}\int_{۰}^{۲\pi} \cos^۲\theta d\theta = \frac{۱}{۴\pi}\int_{۰}^{۲\pi} (۱ + \cos ۲\theta) d\theta = \frac{۲\pi}{۴\pi} = \frac{۱}{۲}$$

$$\Rightarrow \mathbb{E}[\sin^۲\theta_i] = \frac{۱}{۲}$$

$$\begin{aligned}
\mathbb{E}[\cos^۴\theta_i] &= \frac{۱}{۲\pi}\int_{۰}^{۲\pi} \cos^۴ d\theta = \frac{۱}{۸\pi}\int_{۰}^{۲\pi} (۱ + \cos ۲\theta)^۲ d\theta \\
&= \frac{۱}{۸\pi}\int_{۰}^{۲\pi} (۱ + \cos^۲ ۲\theta + ۲\cos ۲\theta) d\theta \\
&= \frac{۱}{۸\pi}\int_{۰}^{۲\pi} (\frac{۳}{۲} + \frac{\cos ۴\theta}{۲} + ۲\cos ۲\theta) d\theta = \frac{۳}{۸} \\
&= \mathbb{E}[\sin^۴\theta_i]
\end{aligned}$$

$$\mathbb{E}[||Z_n||^4] = r^4(n\frac{3}{8} + \frac{n(n+1)}{2}\frac{6}{4}) \times 2 + 2 \times (n\frac{1}{8} + \frac{1}{4}\frac{n(n-1)}{2})$$

$$\begin{aligned}\mathbb{E}[||Z_n||^4] &= r^4(n + \frac{7}{4}n(n-1)) \\ &= \frac{nr^4}{4}(7n-3)\end{aligned}$$

$$||Z_n||^4 = r^4((\sum_{i=1}^{n}\cos\theta_i)^4 + (\sum_{i=1}^{n}\sin\theta_i)^4 + 2(\sum_{i=1}^{n}\cos\theta_i)^2(\sum_{i=1}^{n}\sin\theta_i)^2)$$

$$\forall i \neq j: \qquad \mathbb{E}[\cos^3\theta_i \cos\theta_j] = 0 = \mathbb{E}[\sin^3\theta_i \sin\theta_j]$$

$$\begin{aligned}\forall i \neq j: \qquad \mathbb{E}[\cos^2\theta_i \cos^2\theta_j] &= \mathbb{E}[\cos^2\theta_i]\mathbb{E}[\cos^2\theta_j] \\ &= \frac{1}{4}\end{aligned}$$

۴ گر از این منزل ویران به سوی خانه روم!

۴-۱ پرسش تئوری ۲۰

احتمال اینکه ابتدا روی مبدا باشد و به آن برگردد، برابر است با اینکه ابتدا یک قدم به چپ رفته و از نقطه -۱ به مبدا برسد یا برعکس از راست همین اتفاق بیوفتد. همچنین از سوال ۱۲ میدانیم:

$$\mathrm{P}_n = \begin{cases} ۱ & \mathrm{p} \leqslant \frac{۱}{۲}، \\ (\frac{q}{p})^n & \mathrm{p} > \frac{۱}{۲}. \end{cases}$$

در این سوال $p = \frac{۱}{۲}$

$$\mathrm{P}\{R\} = pP_{۱} + qP_{-۱} = p + q = ۱$$

همچنین از سوال ۲۶ میدانیم:

$$\mathbb{E}[N] = \sum_{i=۱}^{\infty}(۲i\mathrm{P}\{Z_{۲i}\})$$

$$\mathrm{P}\{Z_{۲i}\} = \binom{۲n}{n}(\frac{۱}{۲})^{۲n}$$

از قضیه ای که در سوال ۲۹ مطرح شد داریم:

$$\frac{۱}{\sqrt{\pi(x+\frac{۱}{۲})}} \leqslant \mathrm{P}\{Z_{۲n}\} \leqslant \frac{۱}{\sqrt{\pi n}}$$

احتمال رسیدن به مبدا متناسب با رادیکال n است. چون رادیکال n در مخرج است پس سری واگراست.

۲-۴ پرسش تئوری ۲۱

$$\mathrm{P}\{X_i = x\} = \begin{cases} \frac{۲}{۳} & \text{x= ۱,} \\ \frac{۱}{۲} & \text{x=−۱,} \\ ۰ & o.w. \end{cases}$$

$$\mathrm{P}\{R\} = \frac{۲}{۳} * P_۱ + \frac{۱}{۳} * P_۱ \longrightarrow \begin{cases} P_۱ = (\frac{\frac{۱}{۳}}{\frac{۲}{۳}})^۱ = \frac{۱}{۲} \\ P_{-۱} = ۱ \end{cases}$$

$$\Longrightarrow \mathrm{P}\{R\} = \frac{۱}{۳} + \frac{۱}{۳} = \frac{۲}{۳}$$

$N \longleftarrow$ تعداد دفعاتی که مبدا را دیده است:

$$\mathrm{P}\{N = k\} = \mathrm{P}^{k-۱}\{R\}(۱ - \mathrm{P}\{R\})$$

$$\mathbb{E}[N] = \sum_{k=۱}^{\infty} k\mathrm{P}\{N = k\} = \sum_{k=۱}^{\infty} k\mathrm{P}^{k-۱}\{R\}(۱ - \mathrm{P}\{R\}) = (۱ - \mathrm{P}\{R\}) \sum_{k=۱}^{\infty} k\mathrm{P}^{k-۱}\{R\}$$

$$k\mathrm{P}^{k-۱}\{R\} = \frac{d}{d\mathrm{P}\{R\}}(\mathrm{P}^k\{R\})$$

$$\mathbb{E}[N] = (۱ - \mathrm{P}\{R\}) \sum_{k=۱}^{\infty} \frac{d}{d\mathrm{P}\{R\}}(\mathrm{P}^k\{R\})$$

$$= (۱-\mathrm{P}\{R\}) \sum_{k=۱}^{\infty} \frac{d}{d\mathrm{P}\{R\}}(\frac{\mathrm{P}\{R\}}{۱ - \mathrm{P}\{R\}}) = (۱-\mathrm{P}\{R\})\frac{۱}{(۱ - \mathrm{P}\{R\})^۲} = \frac{۱}{۱ - \mathrm{P}\{R\}} = \frac{۱}{۱ - \frac{۲}{۳}} = ۳$$

۴-۳ پرسش تئوری ۲۲

به تعبیری باید ثابت کنیم که $\mathbb{E}[N] = \infty$
طبق اثباتی که در سوال ۲۹ انجام شده، داریم:

$$\mathbb{E}[N] = \sum_{n=0}^{\infty} \mathrm{P}\{Z_{2n} = 0\}$$

$$\mathrm{P}\{Z_{2n} = 0\} \sim \frac{c}{\sqrt{n}}$$

اگر $M \longrightarrow \infty$:

$$\Longrightarrow \mathbb{E}[N] = \infty$$

$\mathbb{E}[N]$ واگرا میشود.

۴-۴ پرسش تئوری ۲۳

احتمال رسیدن به سر چپ بازه قبل از سر راست قضیه را اینجوری بیان میکنیم که این حرکت هم ارز است با اینکه یک گام به راست برویم و از آنجا ادامه دهیم یا یک گام به چپ برویم و از آنجا ادامه دهیم.
اگر یک گام به سمت راست برویم، انگار محورمان را یک واحد شیفت به چپ داده ایم.در نتیجه:

$$f(x) = pf(x+۱) + qf(x-۱) = \frac{۱}{۲}[f(x+۱) + f(x-۱)] \begin{cases} f(-۱) = ۱ \\ f(M) = ۰ \end{cases}$$

معادله دیفرانسیل این سوال مثل سوال ۱۱ است که حل کردیم، پس در نتیجه:

$$f(x) = A + Bx$$

با اعمال شرایط مرزی، داریم:

$$\begin{cases} f(-۱) = ۱ = A - N \\ f(M) = ۰ = A + MB \end{cases} \implies A = \frac{M}{M+۱}, B = \frac{-۱}{M+۱}$$

$$\implies f(x) = \frac{M-x}{M+۱}$$

۴-۵ پرسش تئوری ۲۴

۱.

$$f(\circ) = \frac{M}{M+۱}$$

۲. از آنجایی که به ازای هر $M \in \mathbb{N}$ برقرار است، پس برای $M \longrightarrow \infty$ نیز برقرار است.

$$\lim_{M\to\infty} f(\circ) = \lim_{M\to\infty} \frac{M}{M+۱} = ۱$$

پس فضا ایستگاه ها را به صورت $[-۱,\infty)$ در نظر بگیرید،در این فضا $f(\circ)$ که همان احتمال رسیدن به -۱ است،برابر با ۱ است.

۳. نتیجه ای که از مسئله قبل میتوانیم بگیریم، این است که در فضای $(-\infty,\infty)$ اگر روی نقطه‌ی π باشیم و با احتمال برابر جا به جا شویم، احتمال رسیدن به $(x-۱)$ $x+۱$ قبل از رسیدن به ∞ $-$ (∞) برابر با ۱ است. پس احتمال بازگشت به مبدا:

$$\mathrm{P}\{R\} = pP_۱ + qP_{-۱} = \frac{۱}{۲}[P_۱ + P_{-۱}] = \frac{۱}{۲}[۱+۱] = ۱$$

پس نتیجه میگیریم که حرکت وفادار است.

۴-۶ پرسش تئوری ۲۵

از صورت قوی قانون اعداد بزرگ داریم:

$$\lim_{n\to\infty} \frac{\sum_{i=1}^{n} X_i}{n} = \mathbb{E}[X]$$

اگر در ابتدا در مبدا باشیم بعد از N حرکت باید به مبدا بازگردیم. اگر حرکت تصادفی باوفا باشد، تعداد N هایی که در معادله زیر صدق میکند، بینهایت است:

$$\sum_{i=1}^{n} X_i = 0$$

در نتیجه از قانون اعداد بزرگ:

$$\lim_{n\to\infty} \frac{\sum_{i=1}^{n} X_i}{N} = 0 \neq \mathbb{E}[X_i]$$

که خلاف فرض صورت سوال است، پس حرکت بی وفا است.

۴-۷ پرسش تئوری ۲۶

تعداد دفعات عبور M_z را میتوان به صورت زیر تعریف کرد:

$$M_z = \sum_{i=0}^{\infty} Y_i$$

که در آن Y_i برابر یک است اگر در حرکت i اُم، روی نقطه‌ی z باشد و در غیر این صورت برابر صفر است.

$$\mathbb{E}[M_z] = \sum_{i=0}^{\infty} \mathrm{P}\{Y_i = 1\} = \sum_{i=0}^{\infty} \mathrm{P}\{Z_i = z\}$$

۴–۸ پرسش تئوری ۲۷

اگر در حرکت n ام به نقطه z برسیم، از آنجا به بعد مسئله رسیدن به نقطه z همان مسئله بازگشت به مبدا است.

$$m(z) = P_z m(0)$$

که در آن P_z احتمال رسیدن از نقطه صفر به نقطه z برای اولین بار است.

$$\forall z \neq 0 : P_z < 1, \forall z = 0 : P_z = 1$$

$$\Longrightarrow \forall z \neq 0 : m(z) = P_z m(0) < m(0)$$

پس نتیجه میگیریم $m(0)$ برای تابع $m(z)$، ماکسیمم است.

۴-۹ پرسش تئوری ۲۸

۱. با استفاده از قضیه مارکف:

$$\begin{cases} \mathrm{P}\{|Z_n| > ۲M\sqrt{n}\} \leqslant \frac{Var(Z_n)}{۴M^۲ n} \\ Var(Z_n) = nVar(X_i) = n\,\mathbb{E}[X_i^۲] \end{cases}$$

$$\longrightarrow \mathrm{P}\{|Z_n| > ۲M\sqrt{n}\} \leqslant \frac{\mathbb{E}[X_i^۲]}{۴M^۲}$$

$$|X_i| < M \Longrightarrow |X_i| < \sqrt{۲}M \longrightarrow X_i^۲ < ۲M^۲$$

$$\mathbb{E}[X_i^۲] < ۲M^۲, \mathrm{P}\{|Z_n| > ۲M\sqrt{n}\} \leqslant \frac{۱}{۲}$$

$$\Longrightarrow \mathrm{P}\{|Z_n| \leqslant ۲M\sqrt{n}\} = ۱ - \mathrm{P}\{|Z_n| > ۲M\sqrt{n}\} \geqslant \frac{۱}{۲}$$

۲. با استفاده از قضیه‌ی سوال ۲۶ و بخش بالایی داریم:

$$\sum_{z=-۲M\sqrt{n}}^{z=۲M\sqrt{n}} m(z) \geqslant \sum_{z=-۲M\sqrt{n}}^{z=۲M\sqrt{n}} \sum_{i=۰}^{n} \mathrm{P}\{Z_i = z\}$$

$$\sum_{z=-۲M\sqrt{n}}^{z=۲M\sqrt{n}} m(z) \geqslant \sum_{i=۰}^{n} \mathrm{P}\{|Z_i| < ۲M\sqrt{n}\} = \sum_{i=۰}^{n} \frac{۱}{۲}$$

$$\sum_{z=-۲M\sqrt{n}}^{z=۲M\sqrt{n}} m(z) \geqslant \frac{۱}{۲}$$

۵ به کوی عشق منه بی دلیل راه، قدم!

۵-۱ پرسش تئوری ۲۹

احتمال بازگشت در مراحل فرد برابر صفر است. احتمال بازگشت در مراحل زوج به صورت زیر است:

$$\mathbb{P}[Z_{2n} = 0] = \binom{2n}{n}\frac{1}{4^n}$$

و از طرفی طبق نامساوی مفروض در صورت سوال

$$\frac{4^n}{\sqrt{\pi(n+\frac{1}{2})}} \leqslant \binom{2n}{n} \leqslant \frac{4^n}{\sqrt{\pi n}}$$

در نتیجه احتمال بین دو کسر با توان منفی یک دوم ساندویچ شده است. پس در مجموع متناسب با آنها خواهد بود. حال کافیست متوسط N را بیابیم:

$$\mathbb{E}[N] = \sum_{n=1}^{\infty} \mathbb{P}[Z_{2n} = 0] \sim \sum_{n=1}^{\infty} \frac{1}{\sqrt{n}} \longrightarrow \infty$$

در حالت کلی نامساوی برای d بعد به شکل زیر در می آید:

$$(\frac{1}{\sqrt{\pi(n+\frac{1}{2})}})^d \leqslant \mathbb{P}[Z_{2n} = 0] \leqslant (\frac{1}{\sqrt{\pi n}})^d$$

که سری متناظر برای محاسبه میانگین دفعات بازگشت به مبدا، سری $p = \frac{-d}{2}$ که به ازای d های بزرگتر از ۲ همگرا و در غیر این صورت واگراست. در نتیجه حرکت در کمتر از ۲ بعد با وفا و در بیش از ۲ بعد بی وفا است.

۵-۲ پرسش تئوری ۳۰

$$Z_n = \sum_{i=۱}^{n} X_i$$

$$X_i \sim \mathcal{N}(۰, \sigma^۲)$$

متغیر های گوسی مستقل، اگر با هم جمع هم بشوند باز هم گوسی اند پس:

$$Z_n \sim \mathcal{N}(۰, n\sigma^۲)$$

$$\Longrightarrow f_{Z_n}(z_n) = \frac{۱}{\sqrt{۲\pi n\sigma^۲}} e^{-\frac{z_n^T z_n}{۲n\sigma^۲}}$$

۵-۳ پرسش تئوری ۳۱

$$f_{Z_{n-1}}(z_n - x_n) = f_{Z_{n-1}}(z_n) + (x_n)^T + \nabla f_{Z_{n-1}}(z_n) + \frac{1}{2}(x_n)^T H_{Z_{n-1}}(z_n)(x_n)$$

$$f_{Z_n}(z_n) = \int_{\mathbb{R}^3} f_{X_n} f_{Z_{n-1}}(z_n - x_n) dx_n$$

$$= \int_{\mathbb{R}^3} f_{X_n} f_{Z_{n-1}}(z_n) dx_n + \int_{\mathbb{R}^3} f_{X_n}(x_n)(x_n)^T \nabla f_{Z_{n-1}}(z_n) dx_n$$

$$+\frac{1}{2}\int_{\mathbb{R}^3} f_{X_n}(x_n)(x_n)^T H_{Z_{n-1}}(z_n)(x_n) dx_n$$

$$\int_{\mathbb{R}^3} f_{X_n} f_{Z_{n-1}}(z_n) dx_n + \frac{1}{2}\int_{\mathbb{R}^3} f_{X_n}(x_n)(x_n)^T H_{Z_{n-1}}(z_n)(x_n) dx_n$$

$$\Longrightarrow f_{Z_n}(z_n) = f_{Z_{n-1}}(z_n) + \frac{1}{2}\int_{\mathbb{R}^3} f_{X_n}(x_n)(x_n)^T H_{Z_{n-1}}(z_n)(x_n) dx_n$$

$$(x_n)^T H_{Z_{n-1}}(z_n)(x_n) = \sum_{j=1}^{n}\sum_{i=1}^{n} x_{n_i}[H_{Z_{n-1}}(z_n)]_{ij} x_{n_i}$$

$$\frac{1}{2}\int_{\mathbb{R}^3} f_{X_n}(x_n)(x_n)^T H_{Z_{n-1}}(z_n)(x_n) dx_n = \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n} x_{n_i}[H_{Z_{n-1}}(z_n)]_{ij}\int_{\mathbb{R}^3} f_{X_n}(x_n) x_{n_i} x_{n_j} dx_n$$

$$\int_{\mathbb{R}^3} f_{X_n}(x_n) x_{n_i} x_{n_j} dx_n = \sigma^2 \delta_{ij}$$

دلتای کرونکر:

$$\delta_{ij} = \begin{cases} 1 & \text{i = j,} \\ 0 & \text{i} \neq \text{j.} \end{cases}$$

$$\frac{1}{2}\int_{\mathbb{R}^3} f_{X_n}(x_n)(x_n)^T H_{Z_{n-1}}(z_n)(x_n) dx_n = \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n} x_{n_i}[H_{Z_{n-1}}(z_n)]_{ij}\sigma^2\delta_{ij}$$

$$= \frac{1}{2}\sigma^2 trace(H_{Z_{n-1}}(z_n))$$

$$f_{Z_n}(z_n) = f_{Z_{n-1}}(z_n) + \frac{1}{2}\sigma^2 trace(H_{Z_{n-1}}(z_n))$$

$$f_{Z_n}(z_n) = f_{Z_{n-1}}(z_n) + \frac{1}{2}\sigma^2 \nabla^2 f_{Z_{n-1}}(z_n)$$

۵-۴ پرسش تئوری ۳۲

$$f_{Z_n}(z_n) = f_{Z_{n-۱}}(z_n) + \frac{۱}{۲}\sigma^۲ trace(H_{Z_{n-۱}}(z_n))$$

$$f_{Z_t}(z_t) = [f_{Z_t}(z_t) - \frac{\partial f_{Z_t}}{\partial t}\delta t] + \frac{۱}{۲}\sigma^۲ trace(H_{Z_{t-۱}}(z_t))$$

$$\frac{\partial f_{Z_t}}{\partial t}\delta t = \frac{۱}{۲}\sigma^۲ trace(H_{Z_{t-۱}}(z_t))$$

$$\frac{\partial f_{Z_t}}{\partial t} = \frac{s}{۲} trace(H_{Z_t}(z_t))$$

$$\alpha = \frac{s}{۲}$$

$$trace(H_{Z_t}(z_t)) = \nabla^۲ f_{Z_t}$$

$$\Longrightarrow \frac{\partial f_{Z_t}}{\partial t} = \frac{s}{۲}\nabla^۲ f_{Z_t}$$

۵-۵ پرسش تئوری ۳۳

$$\tilde{F}(k) \triangleq \mathcal{F}\{f_{Z_t}(z_t)\}$$

$$\frac{d\tilde{F}(k)}{dt} = -\frac{s}{۲}\tilde{F(k)}, k^۲ = k^T k$$

$$\tilde{F}(k) = Ae^{-\frac{sk^T k}{۲}t}$$

برای $t = ۰$ مکان را دقیقا میدانیم کجاست پس: $A = ۱$

$$f_{Z_t}(z_t) = (\frac{۱}{۲\pi})^۳ \int_{\mathbb{R}^۳} e^{-\frac{sk^T k}{۲}t} e^{jk^T x} d^۳k =$$

$$(\frac{۱}{۲\pi})^۳ \int_{-\infty}^{\infty} e^{jk_x x} e^{-\frac{st}{۲}k_x^۲} dk_x \int_{-\infty}^{\infty} e^{jk_y y} e^{-\frac{st}{۲}k_y^۲} dk_y \int_{-\infty}^{\infty} e^{jk_z z} e^{-\frac{st}{۲}k_z^۲} dk_z$$

$$f(t) = \frac{۱}{\sqrt{۲\pi}\sigma} e^{-\frac{t^۲}{۲\sigma^۲}} \Longleftrightarrow g(\omega) = \frac{۱}{\sqrt{۲\pi}} e^{-\frac{\sigma^۲\omega^۲}{۲}}$$

$$f_{Z_t}(z_t) = (\frac{۱}{۲\pi st})^{\frac{۳}{۲}} e^{-\frac{x^۲+y^۲+z^۲}{۲st}}$$

۶-۵ پرسش تئوری ۳۴

$$f_{Z_t}(r) = \int_{\phi=0}^{2\pi}\int_{\theta=0}^{\pi}(\frac{1}{2\pi st})^{\frac{3}{2}}e^{-\frac{r^2}{2st}}r^2\sin\theta d\theta d\phi = 4\pi r^2(\frac{1}{2\pi st})^{\frac{3}{2}}e^{-\frac{r^2}{2st}}$$

$$\mathbb{E}[||Z_t||] = \int_0^\infty r\frac{4\pi r^2}{(2\pi st)^{\frac{3}{2}}}e^{-\frac{r^2}{2st}}dr = \frac{4\pi}{(2\pi st)^{\frac{3}{2}}}\int_0^\infty r^3e^{-\frac{r^2}{2st}}dr$$

حال یک تغییر متغیر میدهیم:

$$\alpha = \frac{r^2}{2st}, d\alpha = \frac{rdr}{st}$$

$$\mathbb{E}[||Z_t||] = \frac{4\pi}{(2\pi st)^{\frac{3}{2}}}\int_0^\infty e^{-\alpha}st(2st\alpha)d\alpha = \frac{2\pi(2st)^2}{(2\pi st)^{\frac{3}{2}}}\int_0^\infty \alpha e^{-\alpha}d\alpha$$

$$\int_0^\infty \alpha e^{-\alpha}d\alpha = \Gamma(2) = 1! = 1$$

$$\Longrightarrow \mathbb{E}[||Z_t||] = \sqrt{\frac{8st}{\pi}}$$

$$\mathbb{E}[||Z_t||^2] = \int_0^\infty r^2\frac{4\pi r^2}{(2\pi st)^{\frac{3}{2}}}e^{-\frac{r^2}{2st}}dr$$

$$\int_0^\infty \frac{4\pi r^2}{(2\pi st)^{\frac{3}{2}}}e^{-\frac{r^2}{2st}}dr = 1, \frac{(2\pi st)^{\frac{3}{2}}}{4\pi} = \int_0^\infty r^2e^{-\frac{r^2}{2st}}dr, \alpha = \frac{1}{2st}$$

$$-\frac{\partial}{\partial\alpha}(\frac{1}{4\sqrt{\pi}}\alpha^{-\frac{3}{2}}) = \int_0^\infty r^4e^{-\alpha r^2}dr$$

$$-\frac{\partial}{\partial\alpha}(\frac{1}{4\sqrt{\pi}}\alpha^{-\frac{3}{2}}) = \frac{3}{2}\frac{1}{4\sqrt{\pi}}\alpha^{-\frac{3}{2}}\frac{1}{\alpha}$$

$$\mathbb{E}[||Z_t||^2] = 3st$$

$$Var[||Z_t||] = 3st - \frac{1}{\pi}(8st) = \frac{3\pi - 8}{\pi}st$$

۷-۵ پرسش تئوری ۳۵

$$f_{Z_t}(z_t) = (\frac{1}{2\pi st})^{\frac{3}{2}} e^{-\frac{x^2+y^2+z^2}{2st}}$$

$$f_{Z_n}(z_n) = (\frac{1}{2\pi n\sigma^2})^{\frac{3}{2}} e^{-\frac{z_n^T z_n}{2n\sigma^2}}$$

اگر با CLT هم برویم به همین را میدهد.

۸-۵ پرسش تئوری ۳۶

$$Z_n = Z_0 \sum_{i=1}^{n} X_i$$

$$\mathbb{E}[Z_n] = 0, Var(Z_n) = \sigma_0^2 + n\sigma^2$$

Z_0 از X_i ها مستقل است پس Z_n گاوسی است:

$$X_i \sim \mathcal{N}(0, \sigma_0^2 + n\sigma^2)$$

$$f_{Z_n} = (\frac{1}{2\pi(\sigma_0^2 + n\sigma^2)})^{\frac{3}{2}} e^{-\frac{Z_n^T Z_n}{2(\sigma_0^2 + n\sigma^2)}}$$

$$f_{Z_t} = (\frac{1}{2\pi(\sigma_0^2 + n\sigma^2)})^{\frac{3}{2}} e^{-\frac{Z_t^T Z_t}{2(\sigma_0^2 + n\sigma^2)}}$$

۹-۵ پرسش تئوری ۳۷

$$f_{Z_n}(z_n) = f^{(}_{X_n}n)(x_n) * f^{(}_{X_{n-1}}n)(z_n - x_n)$$

$$f_{Z_n}(z_n) = f^{(}_{X_1}1)(x_1) * f^{(}_{X_2}2)(x_2) * f^{(}_{X_3}3)(x_3) * ... * f^{(}_{X_n}n)(x_n)$$

$$f_{Z_n}(z_n) = \mathcal{F}^{-1}(\prod_{m=1}^{n} f^{(m)}_{X_m}(\vec{k})) = (\frac{1}{2\pi})^n \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\prod_{m=1}^{\infty} f^{(m)}_{X_m}(\vec{k})d^3k$$

۵-۱۰ پرسش تئوری ۳۸

$$Z_n \sim \mathcal{N}(0, \sigma_0^2 \sum_{m=1}^{n} m^2)$$

$$Z_n \sim \mathcal{N}(0, \sigma_0^2 \frac{n(n+1)(2n+1}{6})$$

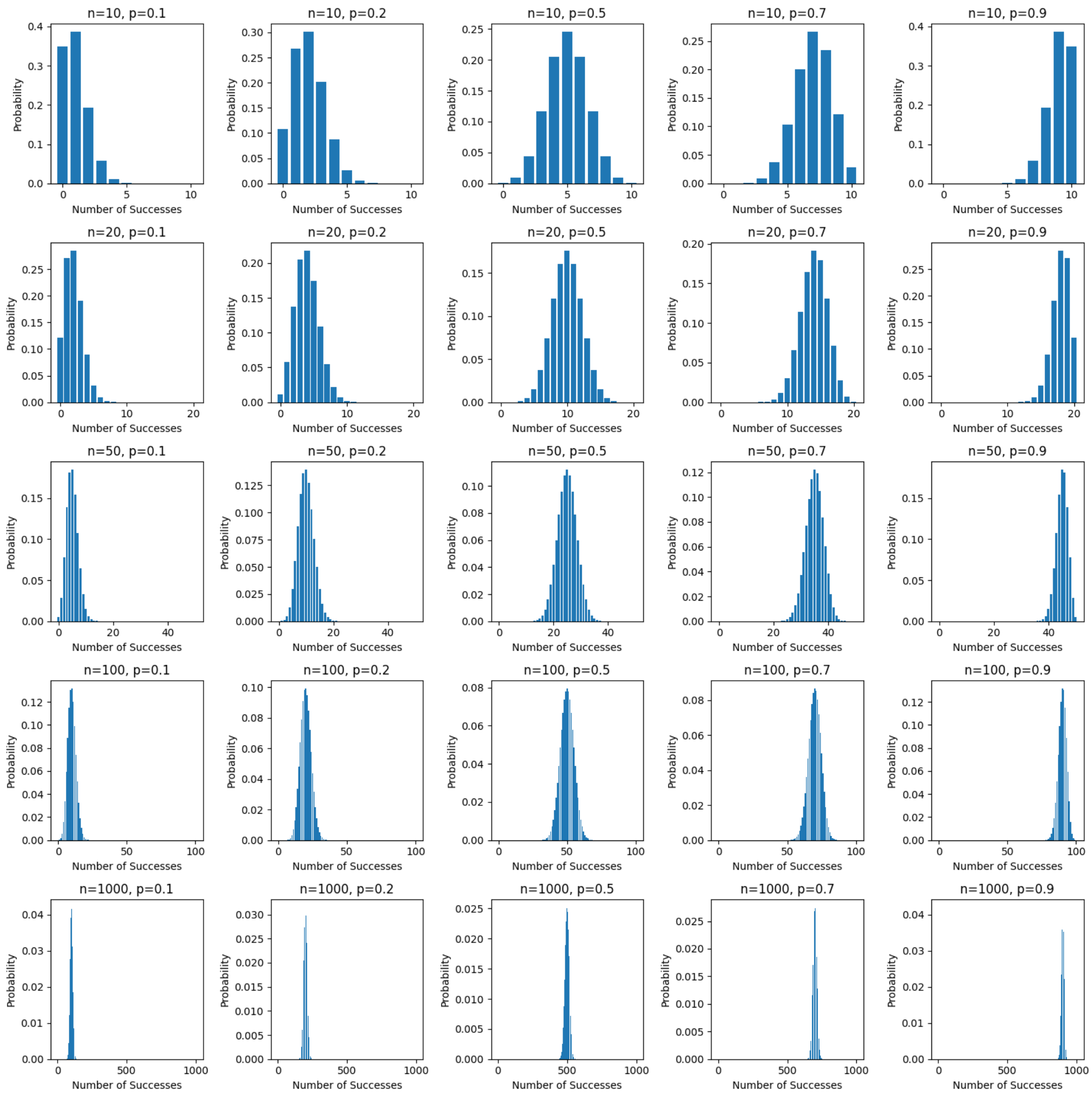
$$f_{Z_n}(z_n) = (\frac{1}{2\pi\sigma_0^2 \frac{n(n+1)(2n+1}{6}})^{\frac{3}{2}} e^{-\frac{z_n^T z_n}{2\sigma_0^2 \frac{n(n+1)(2n+1}{6}}}$$

StaticsProject

February 3, 2024

Question 1

[2]:
```python
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as st

probs = [0.1, 0.2, 0.5, 0.7, 0.9]
nums = [10, 20, 50, 100, 1000]
#here we create a 5*5 figure to plot PMFs
fig, axes = plt.subplots(5, 5, figsize=(15, 15))

#here we just loop through probs and nums
for i, n in enumerate(nums):
    for j, p in enumerate(probs):
        #x-axis
        x = np.arange(0, n+1)
        pmf_values = st.binom.pmf(x, n, p)
        ax = axes[i, j]
        ax.bar(x, pmf_values, align='center')
        ax.set_title(f'n={n}, p={p}')
        ax.set_xlabel('Number of Successes')
        ax.set_ylabel('Probability')

plt.tight_layout()
plt.show()
```



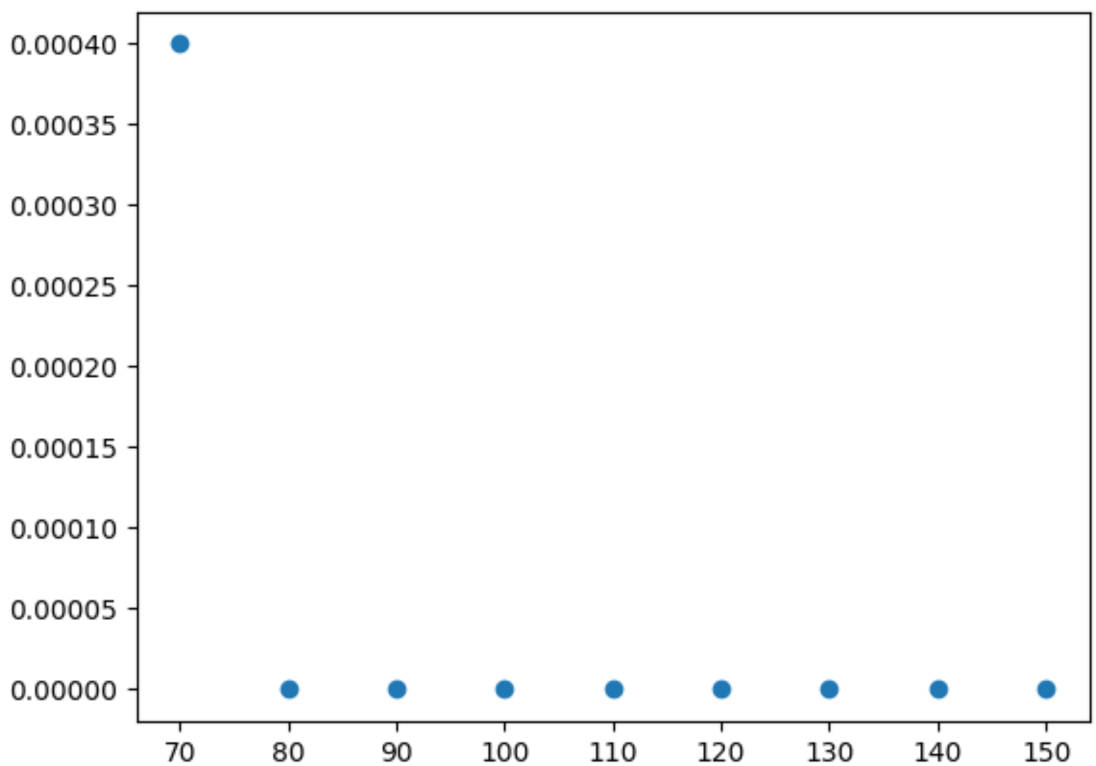
Question 2

[40]:
```
import random
#in this question we use random library to create random nums between 1 and 10
#then we sum all these random nums
count = 0
for i in range(10000):
    sum = 0
    for j in range(79):
        sum += random.randint(1, 10)
    if sum < 301:
        count+=1
```

```
float(count)/10000
```

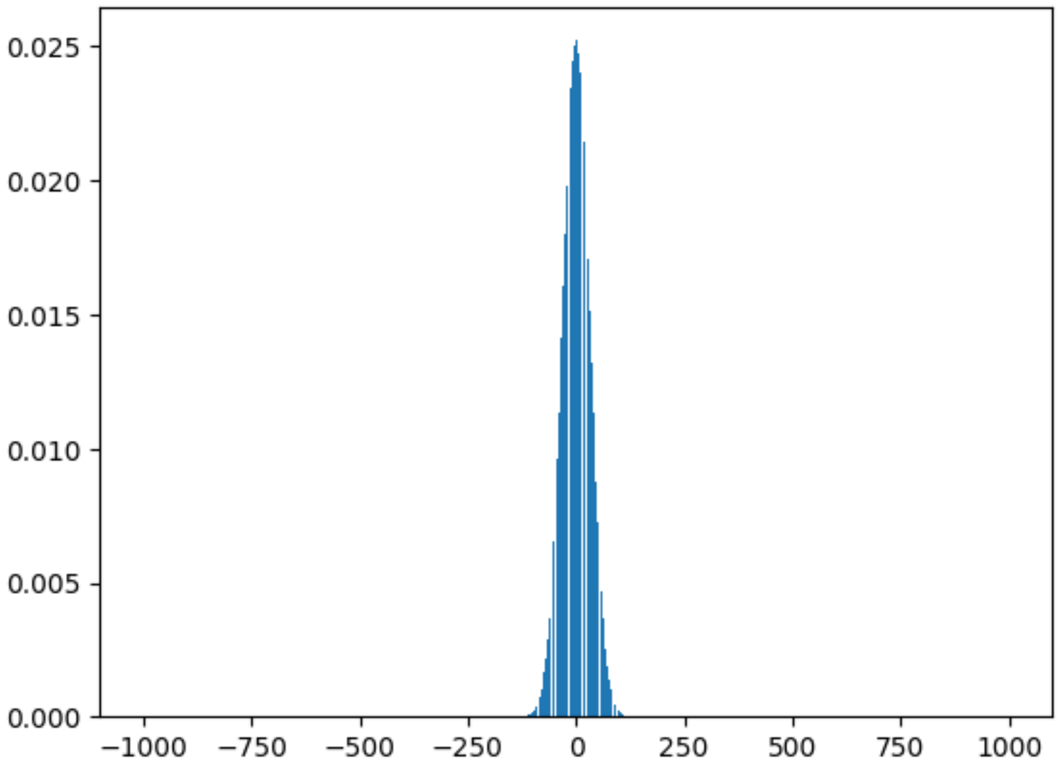
[40]: 0.0

[41]:
```
#here we do the same thing we did in previous part
probs = []
for j in range(70,151,10):
    count = 0
    for i in range(10000):
        sum = 0
        for k in range(j-1):
            sum += random.randint(1,10)
        if sum < 301:
            count += 1
    probs.append(float(count)/10000)
x = list(range(70,151,10))
plt.scatter(x, probs)
plt.show()
```

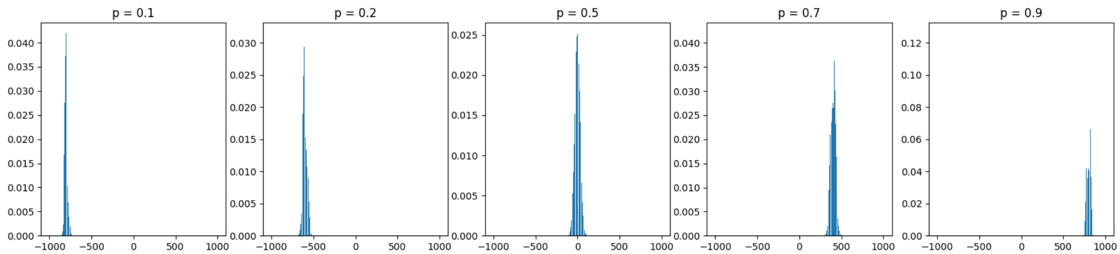


Question 3

[30]:
```
import scipy.special as spc
#here to plot PMF we use scipy.special.comb()
z_vals = np.arange(-1000,1001)
n = 1000
pmf = [spc.comb(n, (n+z)//2) * (0.5**n) for z in z_vals]
plt.bar(z_vals, pmf, align='center')
plt.show()
```



[35]:
```
probs = [0.1, 0.2, 0.5, 0.7, 0.9]
fig, axes = plt.subplots(1, 5, figsize=(20, 4))
#here we plot the PMF for different values of p
for i in range(5):
    pmf_values = [spc.comb(n, (n+z)//2) * probs[i]**((n+z)/2) *␣
 ↪(1-probs[i])**((n-z)/2) for z in z_vals]
    ax = axes[i]
    ax.bar(z_vals, pmf_values, align='center')
    ax.set_title(f'p = {probs[i]}')
```



[37]:
```python
#here we define a function to find the Mean position after 10000 steps
def mean(N,prob):
    pos = []
    for i in range(N):
        pos.append(np.sum(np.random.choice([-1,1], size=10000, p=[1-prob,␣
↪prob])))
    return np.mean(pos)
N = 10000
for p in probs:
    print(f'Mean position after 10000 steps with p = {p} is {mean(N,p)}')
```

```
Mean position after 10000 step with p = 0.1 is -7999.6902
Mean position after 10000 step with p = 0.2 is -5999.0114
Mean position after 10000 step with p = 0.5 is -1.684
Mean position after 10000 step with p = 0.7 is 4002.089
Mean position after 10000 step with p = 0.9 is 8000.0512
```

[38]:
```python
#now we increase the value of N
N = 100000
for p in probs:
    print(f'Mean position after 10000 step with p = {p} is {mean(N,p)}')
```

```
Mean position after 10000 step with p = 0.1 is -7999.93482
Mean position after 10000 step with p = 0.2 is -6000.25278
Mean position after 10000 step with p = 0.5 is -0.36236
Mean position after 10000 step with p = 0.7 is 3999.58712
Mean position after 10000 step with p = 0.9 is 8000.01598
```

Question 4

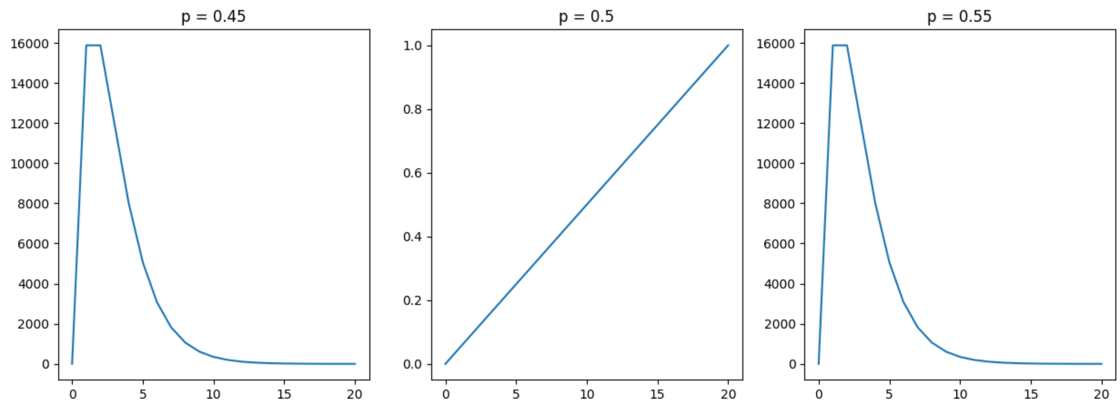
[43]:
```python
#here we define a function to find whether we stop in zoo or in park
#for the part that we want to pick something out of a sample space with specifc␣
↪probabilities
#we use random.choice()
def stop_in_zoo_or_park(N):
    n_zoo = 0
    n_park = 0
    for i in range(N):
```

```python
        pos = 0
        while True:
            x = np.random.choice([-1, 1], p=[0.5, 0.5])
            pos += x
            if pos == 2:
                n_zoo+=1
                break
            if pos == -1:
                n_park+=1
                break
    return [n_zoo, n_park]

ans = stop_in_zoo_or_park(100000)
print('Number of times stoped in zoo: ', ans[0])
print('Number of times stoped in park: ', ans[-1])
```

```
Number of times stoped in zoo:  33513
Number of times stoped in park:  66487
```

Question 5

[46]:
```python
def exit(l, probs):
    n = np.arange(0,l+1)

    fig, axes = plt.subplots(1, 3, figsize=(15, 5))

    for i in range(3):
        ax = axes[i]
        ax.set_title(f'p = {probs[i]}')
        if(probs[i] == 0.5):
            P = n / l
            ax.plot(n, P)
        else:
            P = (probs[i]**n - (1-probs[i])**n) / (probs[i]**l - (1-probs[i])**l)
            ax.plot(n, P)

probs = [0.45, 0.5, 0.55]
exit(20, probs)
```



Question 6

[86]:
```python
def number_of_times_passing(k,n,prob):
    count = 0
    pos = 0
    for i in range(n):
        pos += np.random.choice([-1, 1], p=[1-prob, prob])
        if pos == k:
            count += 1
    return count
count = []
for i in range(1000):
    count.append(number_of_times_passing(2, 10000, 0.3))
print(f'Mean number of times passing from +2 after 10000 steps is ', np.
 ↪mean(count))
```

```
Mean number of times passing from +2 after 10000 steps is  0.422
```

Question 7

[82]:
```python
def time_reaching_position(k, prob):
    pos = 0
    time = 0
    while True:
        time += 1
        pos += np.random.choice([-1, 1], p=[1-prob, prob])
        if pos == k:
            break
    return time
time = []
for i in range(10000):
    time.append(time_reaching_position(50, 0.55))
print(f'Mean time reaching position k is', np.mean(time))
```

```
Mean time reaching position k is 502.9958
```

Question 8

[45]:
```
n = [10, 20, 50, 100, 200]
N = 10000
z0 = 100
count = []

for i in n:
    c = 0
    for j in range(N):
        pos = z0
        pos += np.sum(np.random.normal(0, 1, size=i))
        if(pos > 105):
            c += 1
    count.append(c)
print(count)
```

```
[536, 1337, 2401, 2993, 3599]
```

Question 9

[46]:
```
count = 0
for i in range(1000):
    pos = 0
    for j in range(100):
        pos += np.random.normal(5, 1)
        if(pos <= 0):
            count += 1
            break
print((1000-count)/1000)
```

```
1.0
```

[48]:
```
count = 0
for i in range(1000):
    pos = 0
    for j in range(100):
        pos += np.random.normal(2, np.sqrt(5))
        if(pos <= 0):
            count += 1
            break
print((1000-count)/1000)
```

```
0.741
```

[50]:
```
count = 0
res = []
for i in range(1000):
    pos = 0
    s = []
```

```python
    for j in range(100):
        a = np.random.normal(5, 1)
        pos += a
        s.append(pos)
    res.append(len(set(s)))
print(np.sum(res)/(100000))
```

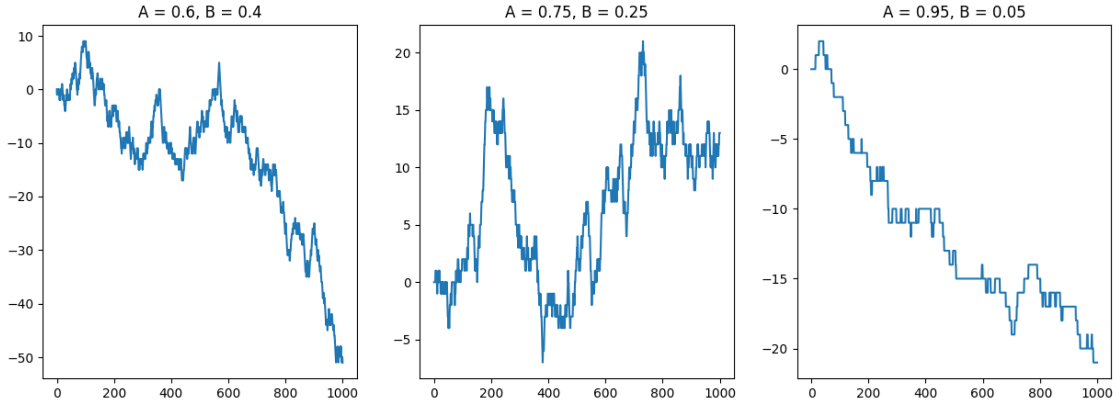
1.0

[51]:
```python
count = 0
res = []
for i in range(1000):
    pos = 0
    s = []
    for j in range(100):
        a = np.random.normal(2, np.sqrt(5))
        pos += a
        s.append(pos)
    res.append(len(set(s)))
print(np.sum(res)/(100000))
```

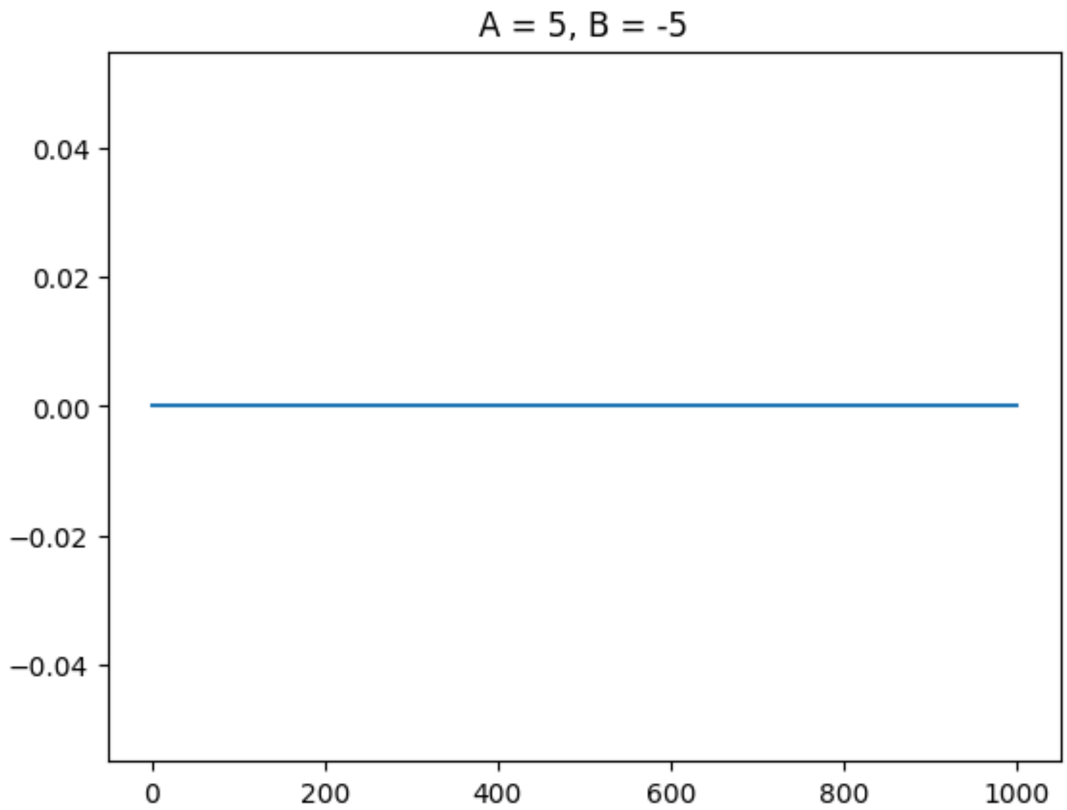
1.0

Question 10

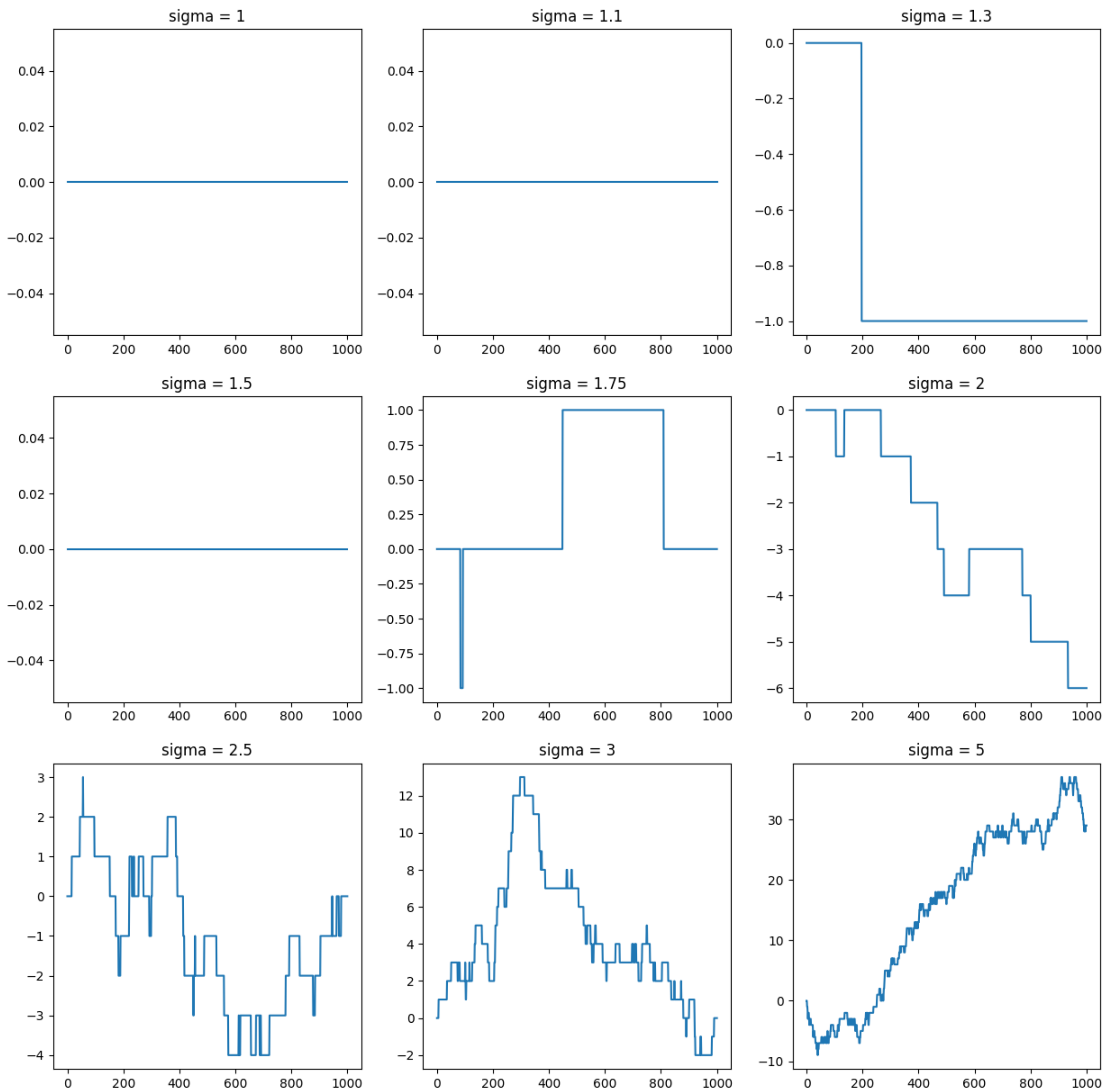
[90]:
```python
def Uniform_move(vals, n):
    a = len(vals)

    fig, axes = plt.subplots(1, a, figsize=(a * 5, 5))

    for i in range(a):
        ax = axes[i]
        ax.set_title(f'A = {vals[i][0]}, B = {vals[i][-1]}')
        pos = np.zeros(n+1)
        for j in range(1,n+1):
            Y = np.random.uniform(0, 1)
            if Y >= vals[i][0]:
                pos[j] = pos[j-1] + 1
            elif Y <= vals[i][-1]:
                pos[j] = pos[j-1] - 1
            else:
                pos[j] = pos[j-1]
        ax.plot(range(n+1), pos)

vals = [[0.6, 0.4], [0.75, 0.25], [0.95, 0.05]]
Uniform_move(vals, 1000)
```



[93]:
```
def Gaussian_move(vals, n, mean, variance):
    a = len(vals)
    for i in range(a):
        plt.title(f'A = {vals[i][0]}, B = {vals[i][-1]}')
        pos = np.zeros(n+1)
        for j in range(1,n+1):
            Y = np.random.normal(mean, np.sqrt(variance))
            if Y >= vals[i][0]:
                pos[j] = pos[j-1] + 1
            elif Y <= vals[i][-1]:
                pos[j] = pos[j-1] - 1
            else:
                pos[j] = pos[j-1]
        plt.plot(range(n+1), pos)

vals = [[5, -5]]
Gaussian_move(vals, 1000, 0, 1)
```

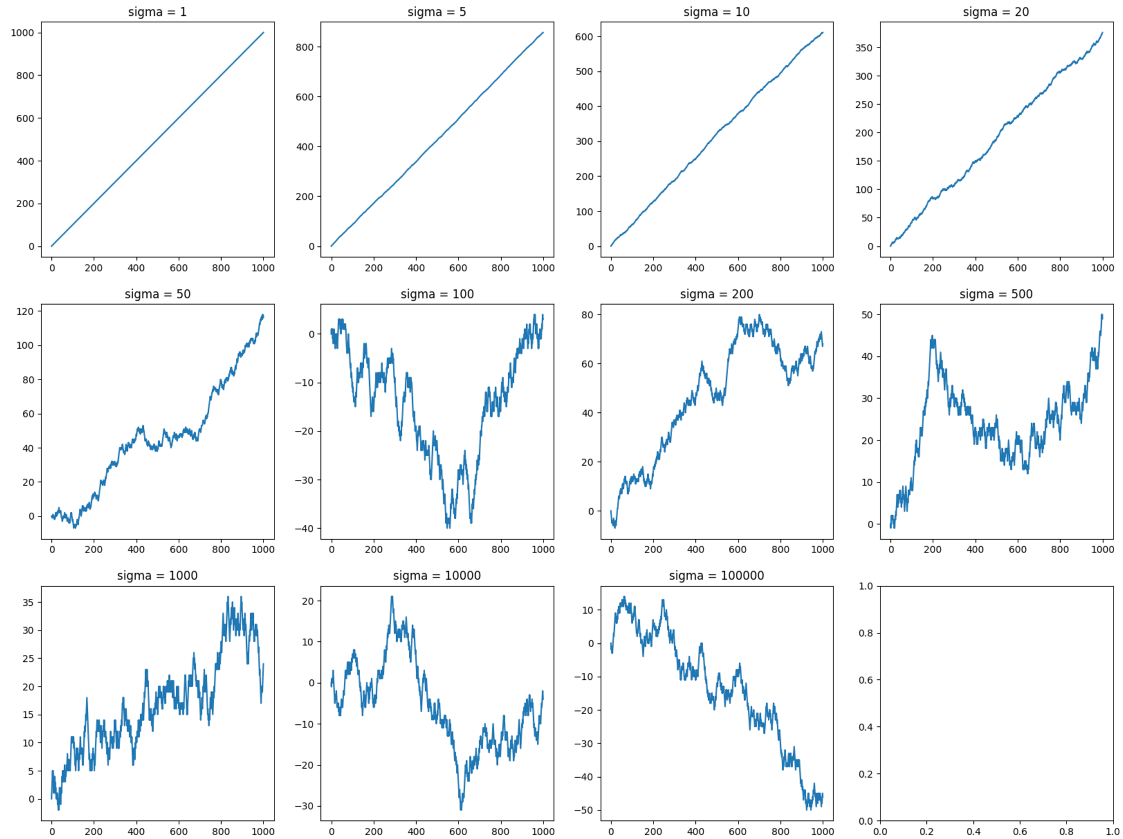


[103]:
```
def Gaussian_move_variance(vars, n, mean):
    a = len(vars)

    fig, axes = plt.subplots(3, a//3, figsize=((a//3) * 5, 15))

    for i in range(a):
        ax = axes[i//3][i%3]
        ax.set_title(f'sigma = {vars[i]}')
        pos = np.zeros(n+1)
        for j in range(1,n+1):
            Y = np.random.normal(mean, vars[i])
            if Y >= 5:
                pos[j] = pos[j-1] + 1
            elif Y <= -5:
                pos[j] = pos[j-1] - 1
            else:
                pos[j] = pos[j-1]
        ax.plot(range(n+1), pos)

vars = [1, 1.1, 1.3, 1.5, 1.75, 2, 2.5, 3, 5]
Gaussian_move_variance(vars, 1000, 0)
```



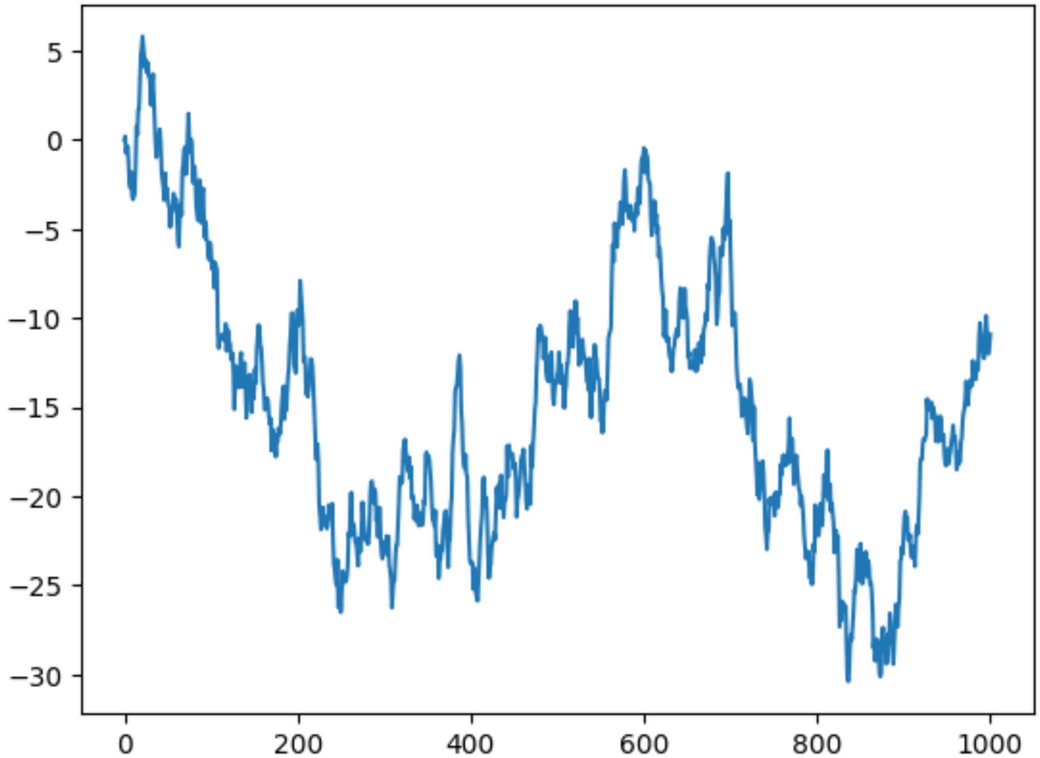
[108]:
```
def Gaussian_move_variance(vars, n, mean):
    a = len(vars)

    fig, axes = plt.subplots(3, a//3 + 1, figsize=((a//3 + 1) * 5, 15))

    for i in range(a):
        ax = axes[i//4][i%4]
        ax.set_title(f'sigma = {vars[i]}')
        pos = np.zeros(n+1)
        for j in range(1,n+1):
            Y = np.random.normal(mean, vars[i])
            if Y >= 5:
                pos[j] = pos[j-1] + 1
```

```
            elif Y <= -5:
                pos[j] = pos[j-1] - 1
            else:
                pos[j] = pos[j-1]
        ax.plot(range(n+1), pos)
vars = [1, 5, 10, 20, 50, 100, 200, 500, 1000, 10000, 100000]
Gaussian_move_variance(vars, 1000, 10)
```

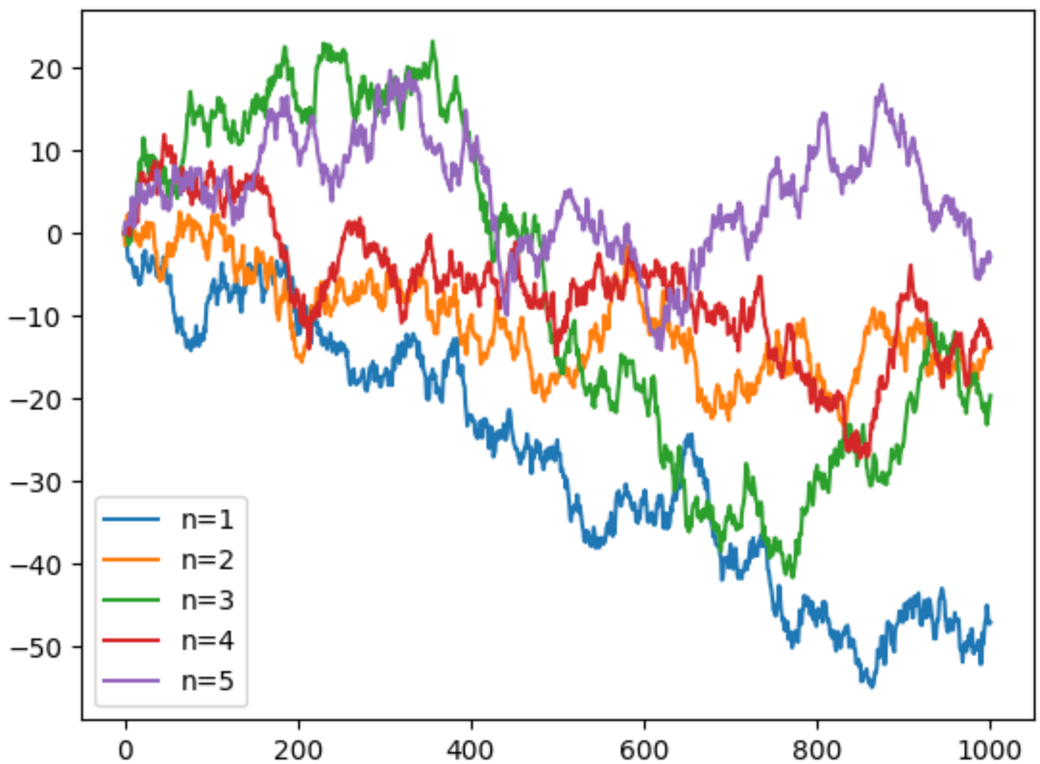


Question 11

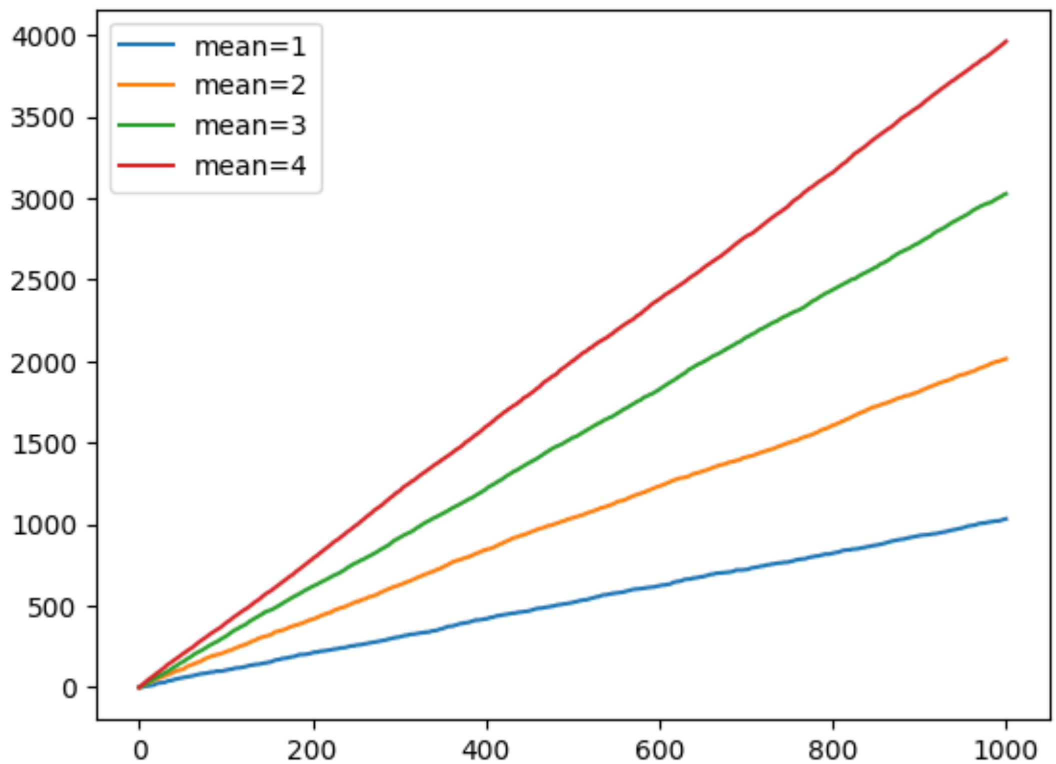
[110]:
```
n = 1000
pos = np.zeros(n+1)
for i in range(1, n+1):
    x = np.random.normal(0, 1)
    pos[i] = pos[i-1] + x

plt.plot(range(n+1), pos)
plt.show()
```



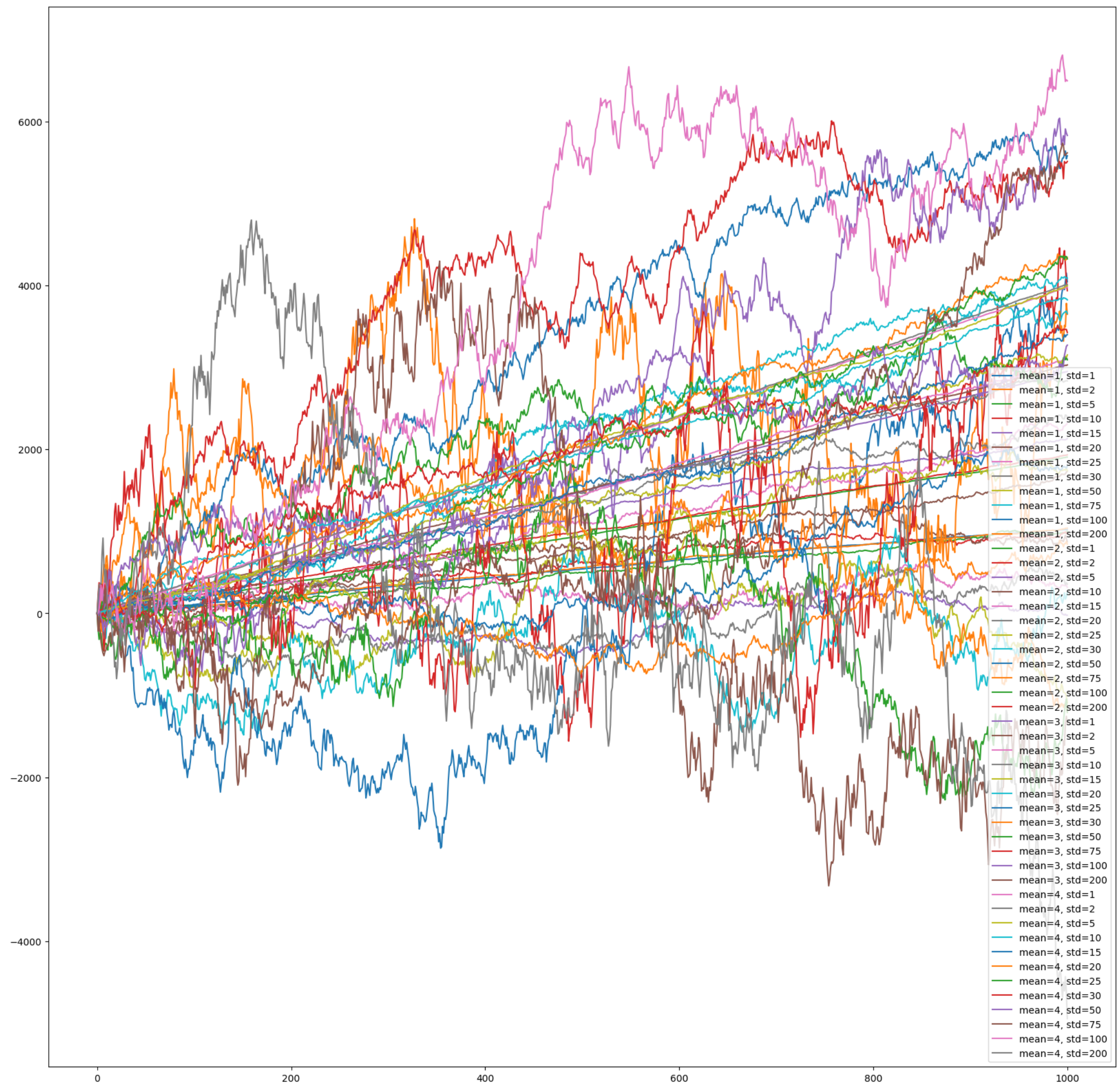
[112]:
```
for i in range(5):
    pos = np.zeros(n+1)
    for j in range(1, n+1):
        x = np.random.normal(0, 1)
        pos[j] = pos[j-1] + x
    plt.plot(range(n+1), pos, label=f'n={i+1}')
plt.legend()
plt.show()
```



[113]:
```
means = [1, 2, 3, 4]
for i in means:
    pos = np.zeros(n+1)
    for j in range(1, n+1):
        x = np.random.normal(i, 1)
        pos[j] = pos[j-1] + x
    plt.plot(range(n+1), pos, label=f'mean={i}')
plt.legend()
plt.show()
```

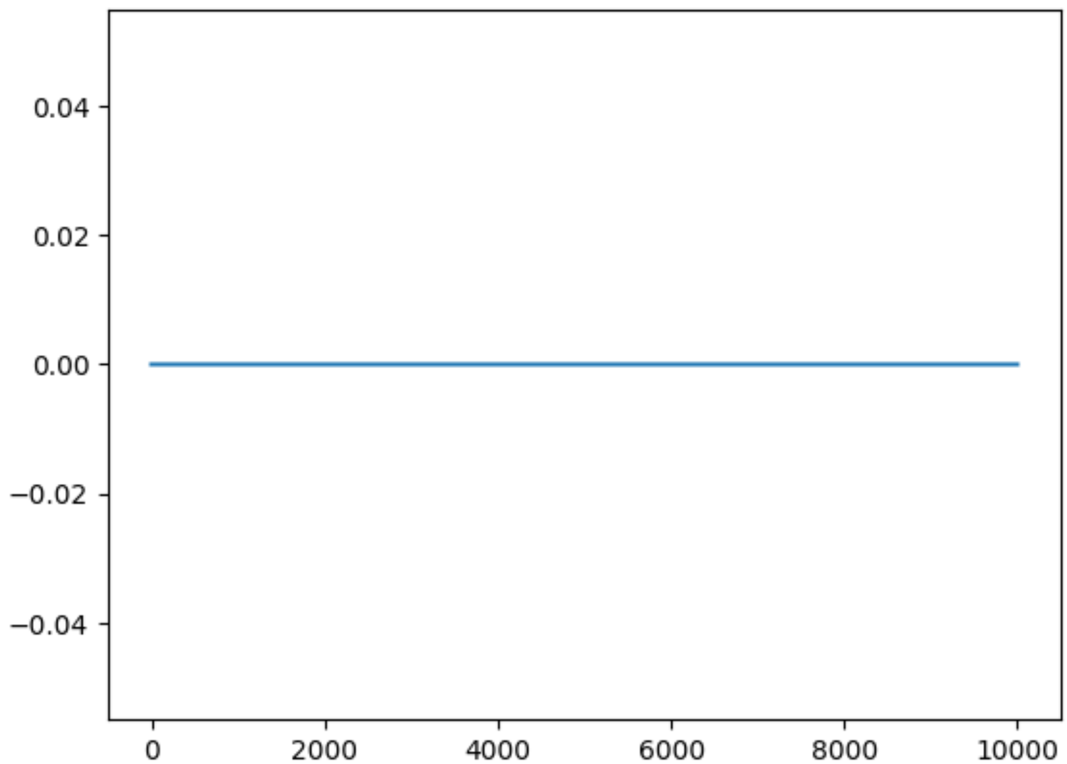


[118]:
```
vars = [1, 2, 5, 10, 15, 20, 25, 30, 50, 75, 100, 200]
plt.figure(figsize=(20, 20))
for m in means:
    for v in vars:
        for j in range(1, n+1):
            x = np.random.normal(m, v)
            pos[j] = pos[j-1] + x
        plt.plot(range(n+1), pos, label=f'mean={m}, std={v}')

plt.legend()
plt.show()
```



Question 12

[121]:
```
n = 10000
pos = np.zeros(n+1)
for i in range(1, n+1):
    x = np.random.gamma(shape=0, scale=1)
    pos[i] = pos[i-1] + x

plt.plot(range(n+1), pos)
plt.show()
```

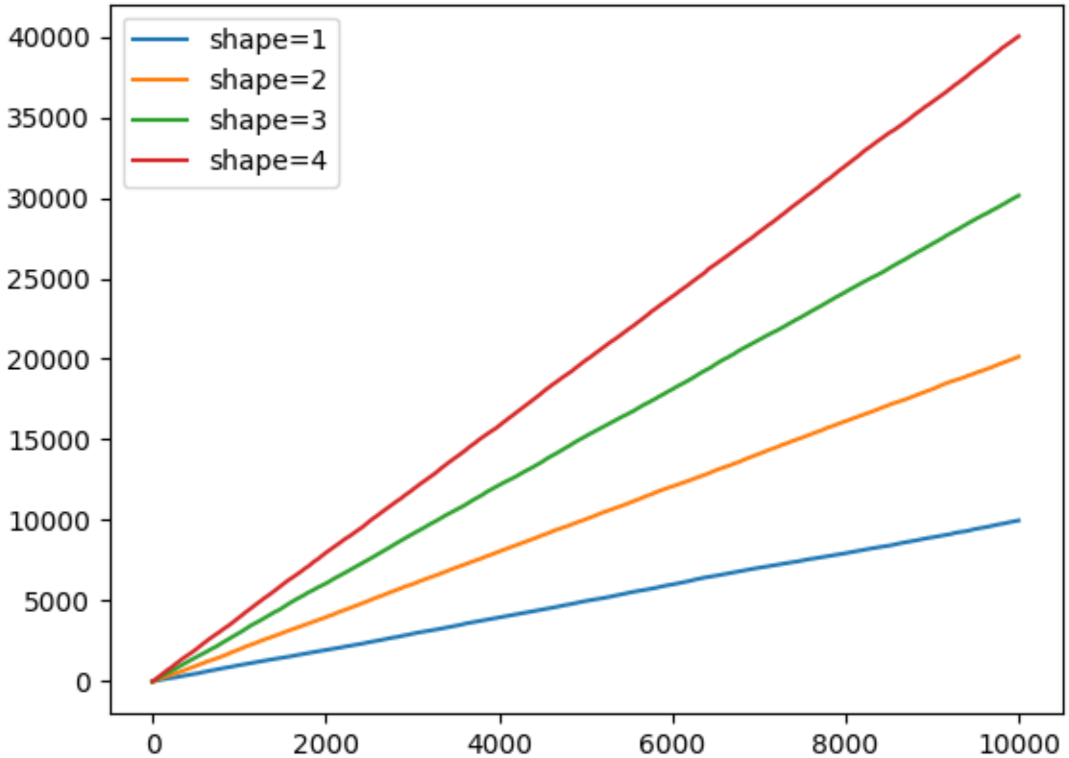


[122]:

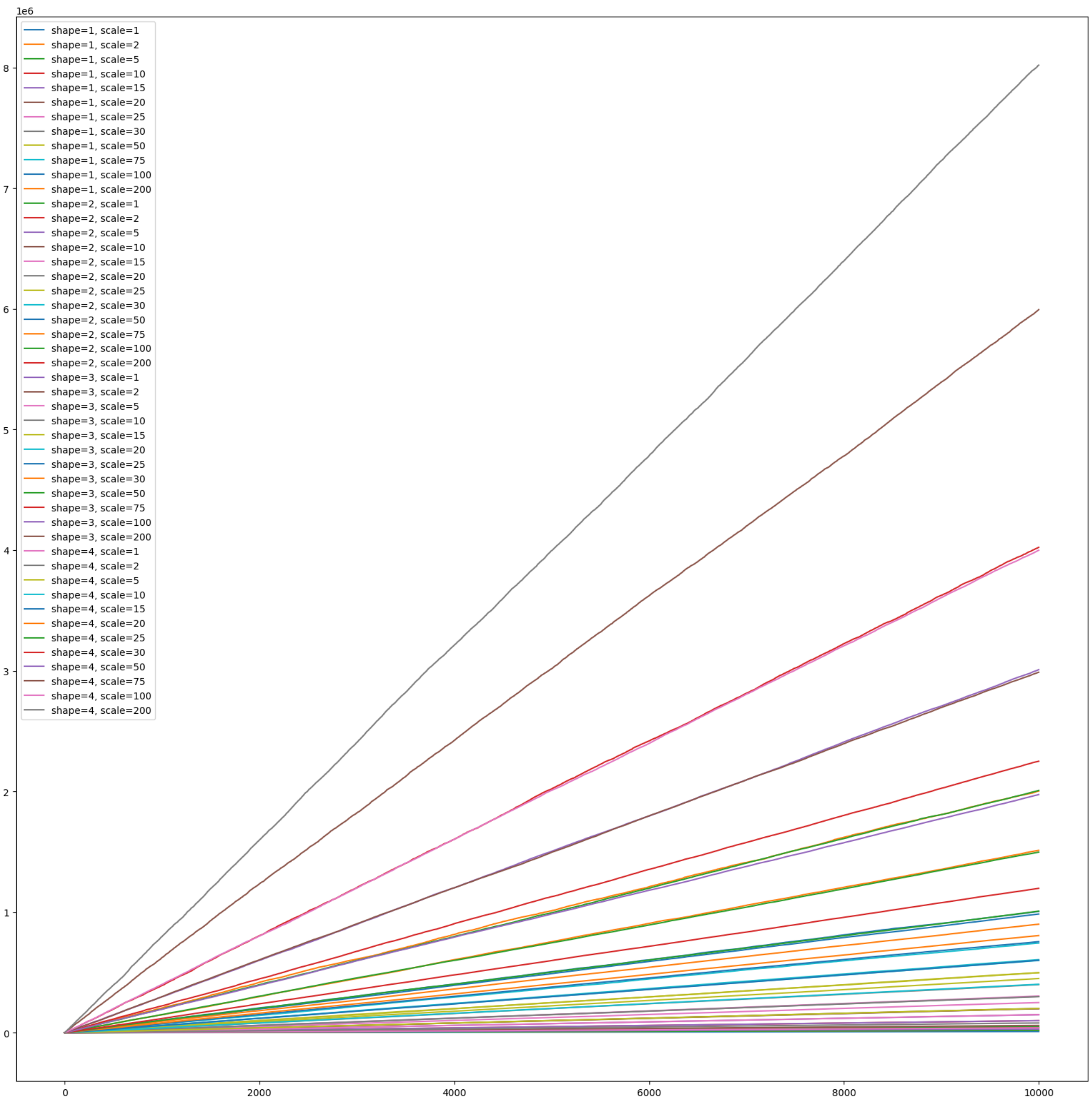
```
#the result is as same as above there is no need to replot the shape above
```

[123]:

```
shapes = [1, 2, 3, 4]
for i in shapes:
    pos = np.zeros(n+1)
    for j in range(1, n+1):
        x = np.random.gamma(shape=i, scale=1)
        pos[j] = pos[j-1] + x
    plt.plot(range(n+1), pos, label=f'shape={i}')
plt.legend()
plt.show()
```



[124]:
```
vars = [1, 2, 5, 10, 15, 20, 25, 30, 50, 75, 100, 200]
plt.figure(figsize=(20, 20))
for m in means:
    for v in vars:
        for j in range(1, n+1):
            x = np.random.gamma(m, v)
            pos[j] = pos[j-1] + x
        plt.plot(range(n+1), pos, label=f'shape={m}, scale={v}')

plt.legend()
plt.show()
```



Question 13

[4]:
```python
def check_origin(n,prob):
    count = 0
    pos = 0
    for i in range(n):
        pos += np.random.choice([1, -1], p=[prob, 1-prob])
        if pos == 0:
            count += 1
    return count
count = []
for i in range(10000):
    count.append(check_origin(100, 0.5))
```

```
print(f'Probability of getting back to origin is ', (len(count) - count.
 ↪count(0))/10000)
print(f'Mean number of times that gets back to origin is ', np.mean(count))
```

```
Probability of getting back to origin is  0.919
Mean number of times that gets back to origin is  7.0434
```

[5]:
```
#here we increase n to see the result
count = []
for i in range(10000):
    count.append(check_origin(1000, 0.5))
print(f'Probability of getting back to origin is ', (len(count) - count.
 ↪count(0))/10000)
print(f'Mean number of times that gets back to origin is ', np.mean(count))
```

```
Probability of getting back to origin is  0.9741
Mean number of times that gets back to origin is  24.5106
```

[6]:
```
#here we just change the value of p
count = []
for i in range(10000):
    count.append(check_origin(100, 0.6666667))
print(f'Probability of getting back to origin is ', (len(count) - count.
 ↪count(0))/10000)
print(f'Mean number of times that gets back to origin is ', np.mean(count))
```

```
Probability of getting back to origin is  0.6661
Mean number of times that gets back to origin is  1.9974
```

[7]:
```
count = []
for i in range(10000):
    count.append(check_origin(200, 0.6666667))
print(f'Probability of getting back to origin is ', (len(count) - count.
 ↪count(0))/10000)
print(f'Mean number of times that gets back to origin is ', np.mean(count))
```

```
Probability of getting back to origin is  0.6554
Mean number of times that gets back to origin is  1.9522
```

Question 14

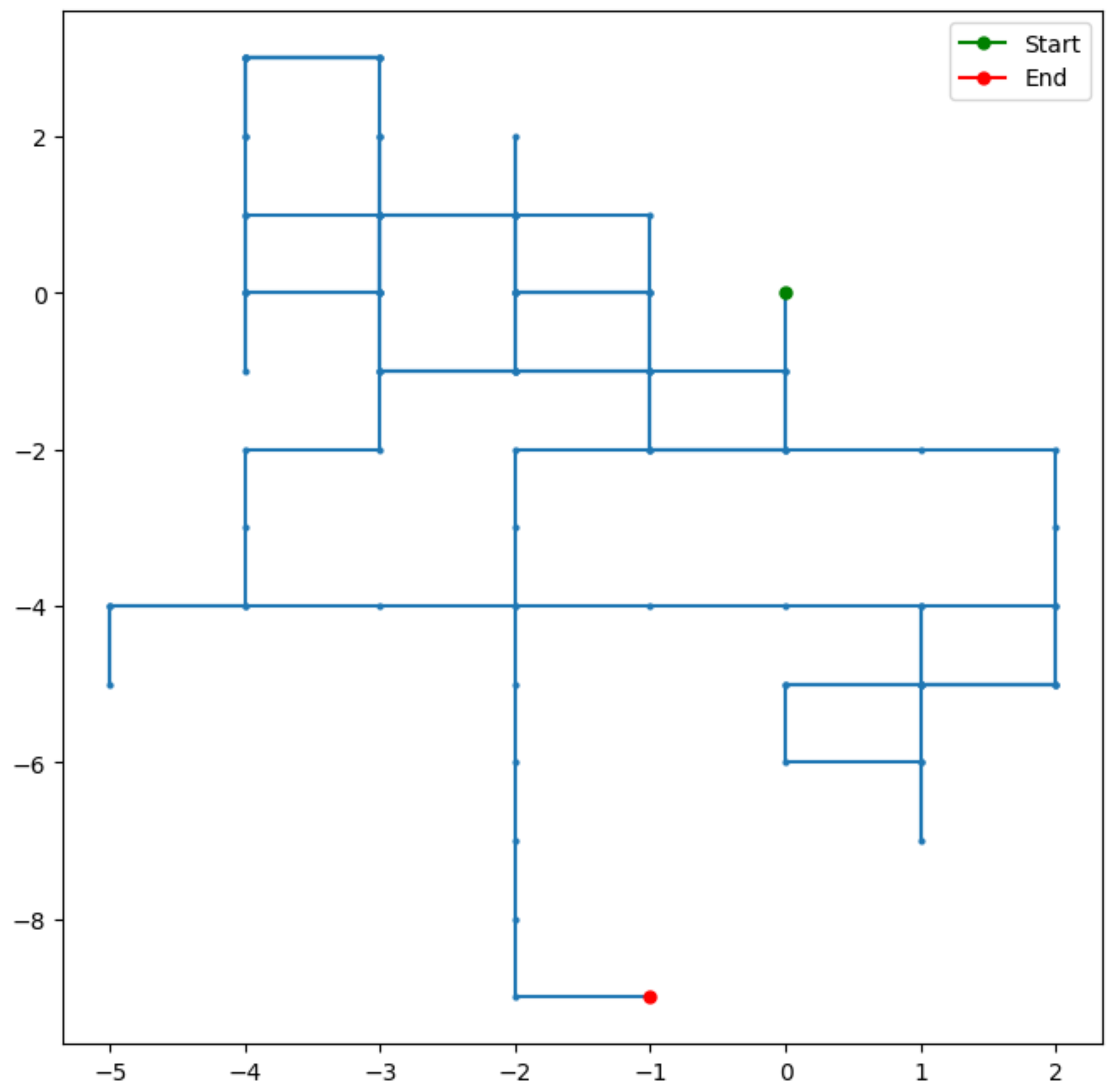
[132]:
```
def Check_if_reached(n,l,h):
    pos = 0
    for i in range(n):
        pos += np.random.choice([1, -1], p=[0.5,0.5])
        if pos == l:
            return True
        elif pos == h:
            return False
```

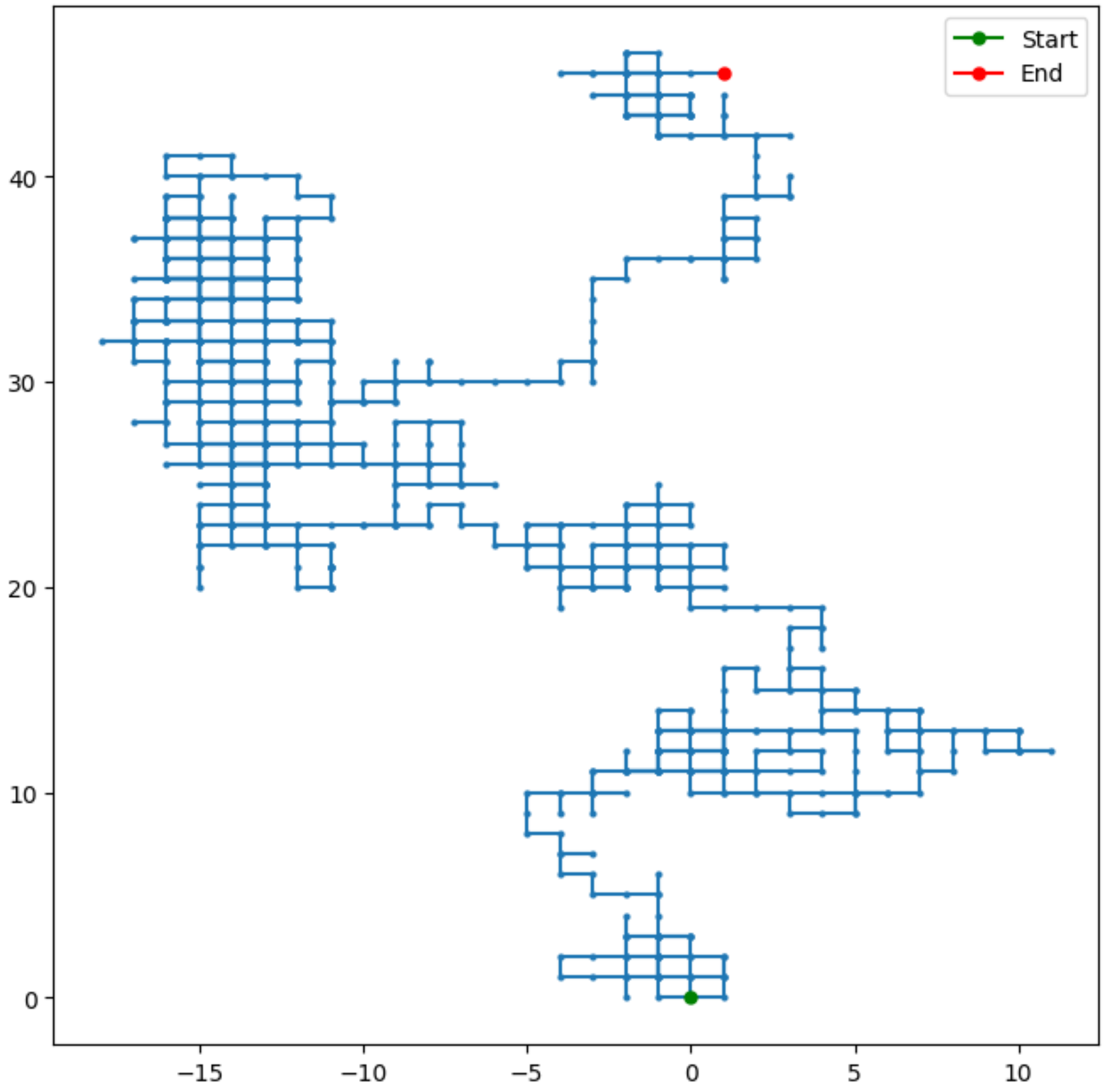
```
for j in range(5,31,5):
    count = 0
    for i in range(1000):
        if Check_if_reached(1000, -1, j):
                count += 1
    print(f'Probability of reaching -1 earlier than {j} is ', count/1000)
```

```
Probability of reaching -1 earlier than 5 is  0.831
Probability of reaching -1 earlier than 10 is  0.91
Probability of reaching -1 earlier than 15 is  0.948
Probability of reaching -1 earlier than 20 is  0.954
Probability of reaching -1 earlier than 25 is  0.96
Probability of reaching -1 earlier than 30 is  0.958
```

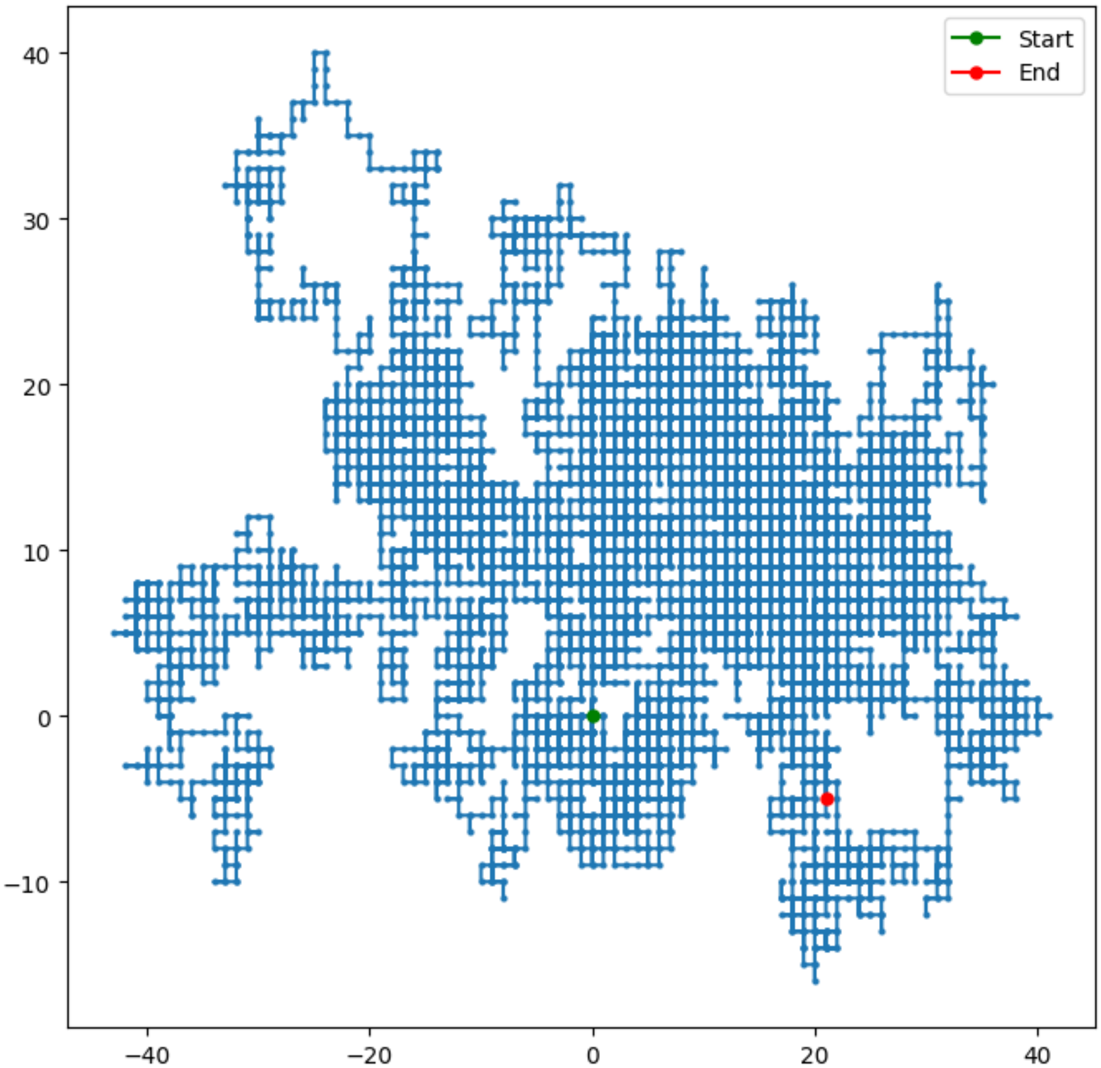
Question 15

[18]:
```
def random_walk_2D(n, dxdy):
    pos = np.zeros((n+1, 2))
    pos[0] = [0, 0]

    for i in range(1, n+1):
        x = dxdy[np.random.choice(len(dxdy), p=[1/len(dxdy) for i in␣
↪range(len(dxdy))])]
        pos[i] = pos[i-1] + x

    plt.figure(figsize=(8, 8))
    plt.plot(pos[:, 0], pos[:, 1], linestyle='-', marker='o', markersize=2)
    plt.plot(pos[0][0], pos[0][1], marker='o', markersize=5, color='green',␣
↪label='Start')
    plt.plot(pos[:, 0][-1], pos[:, 1][-1], marker='o', markersize=5,␣
↪color='red', label='End')
    plt.legend()

n = [10**i for i in range(2, 6)]
dxdy = np.array([[1, 0], [-1, 0], [0, 1], [0, -1]])
for i in n:
    random_walk_2D(i, dxdy)
```

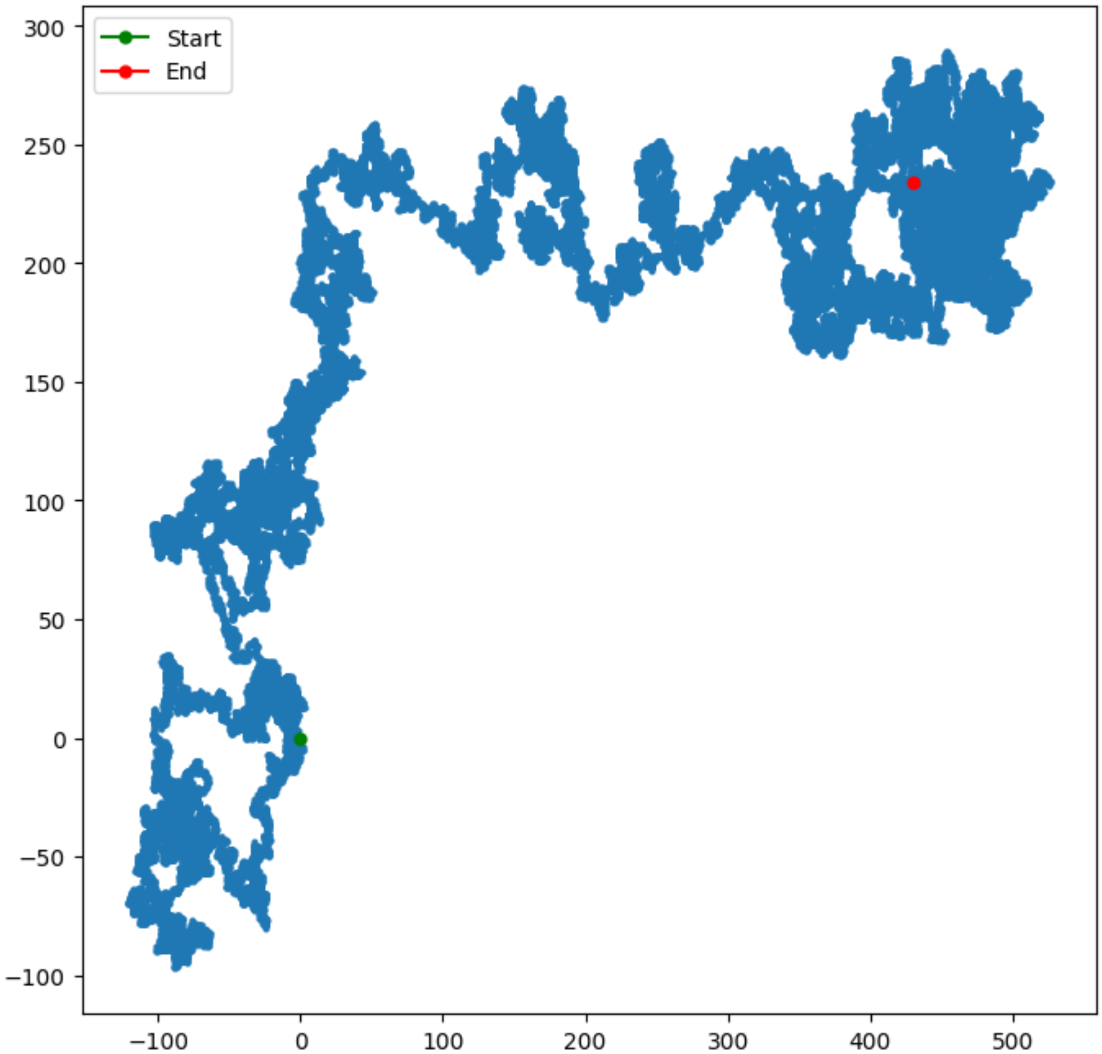

Start
End
2
0
−2
−4
−6
−8
−5
−4
−3
−2
−1
0
1
2



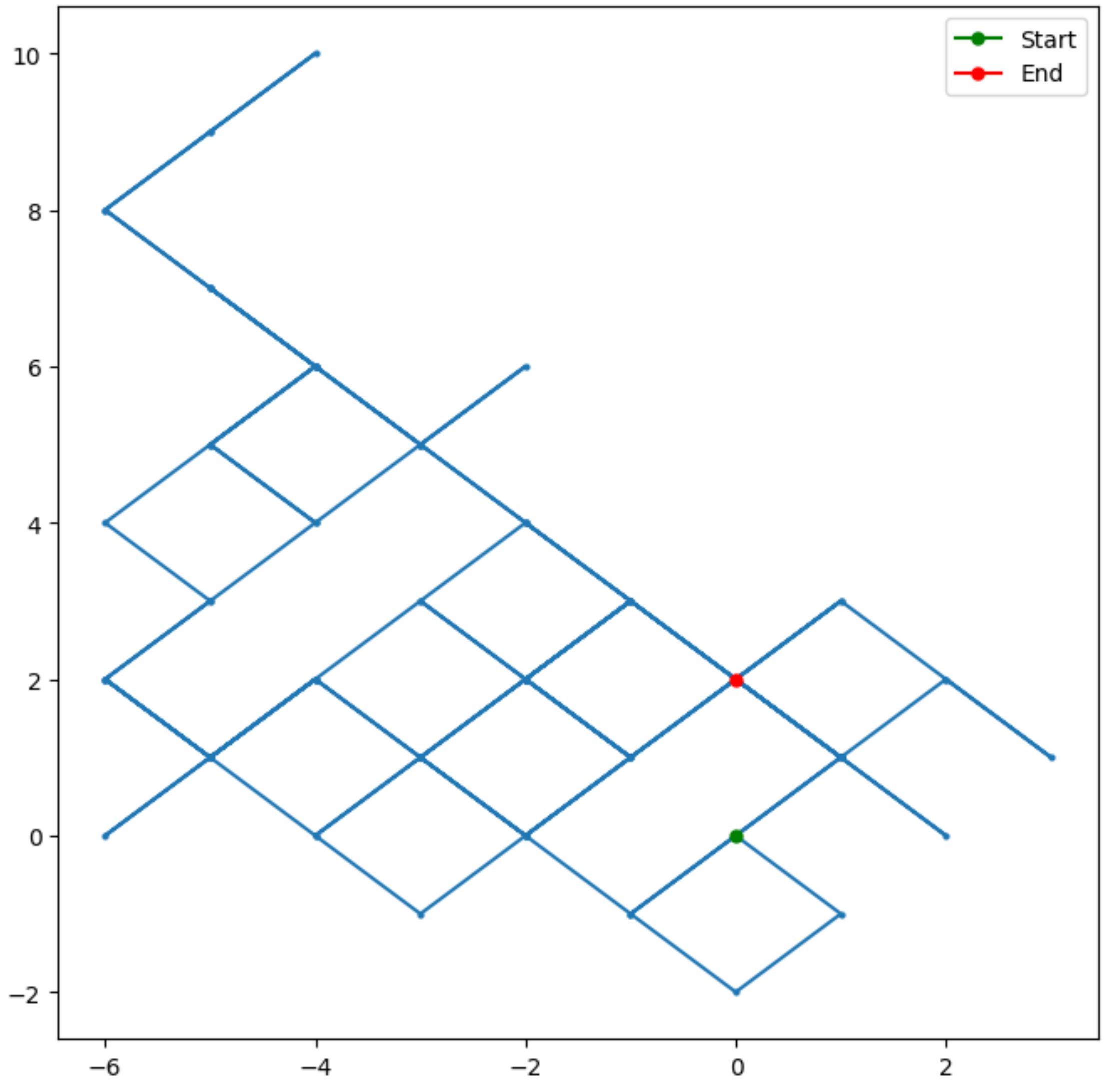
Start
End
40
30
20
10
0
−15
−10
−5
0
5
10

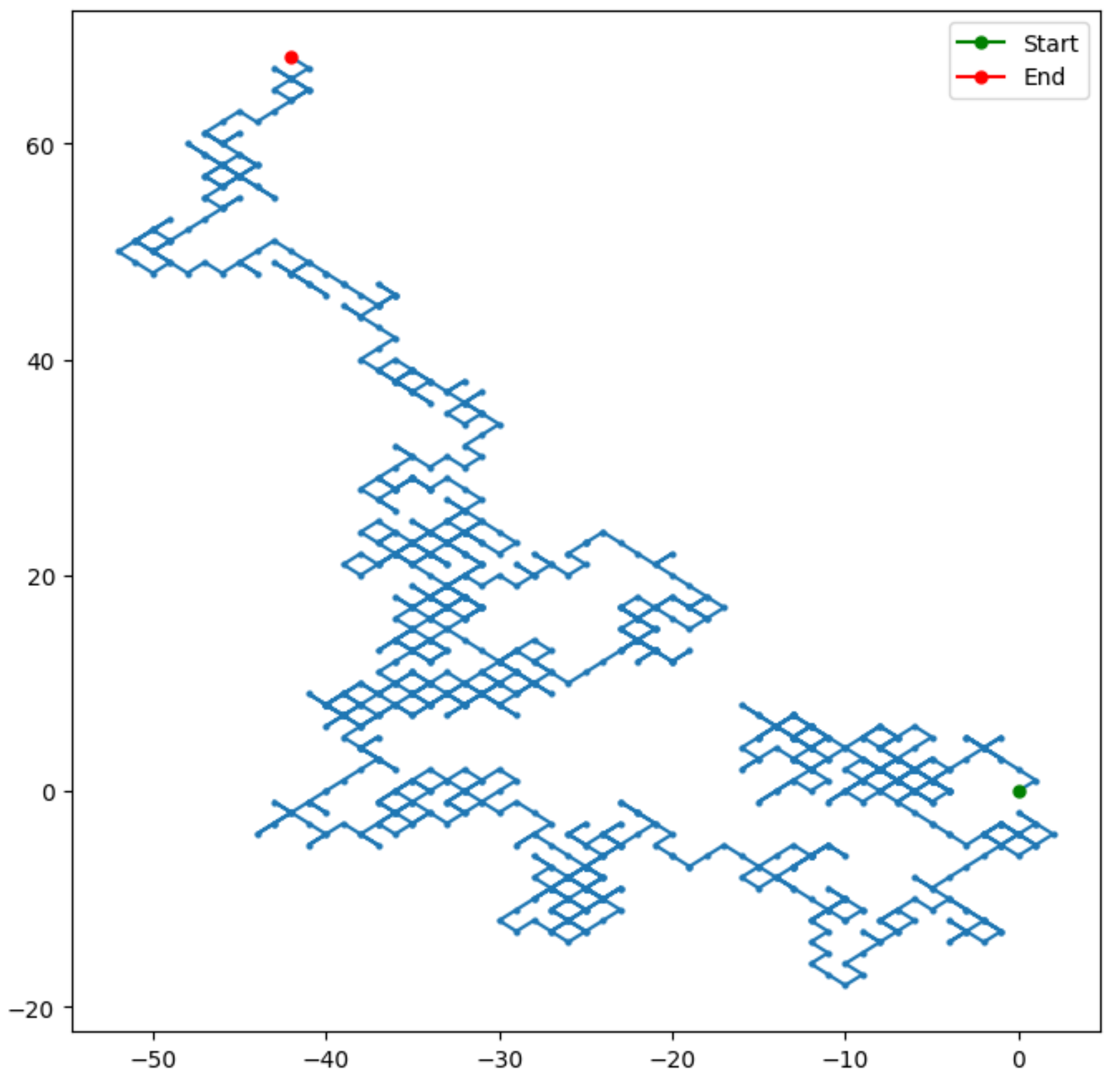


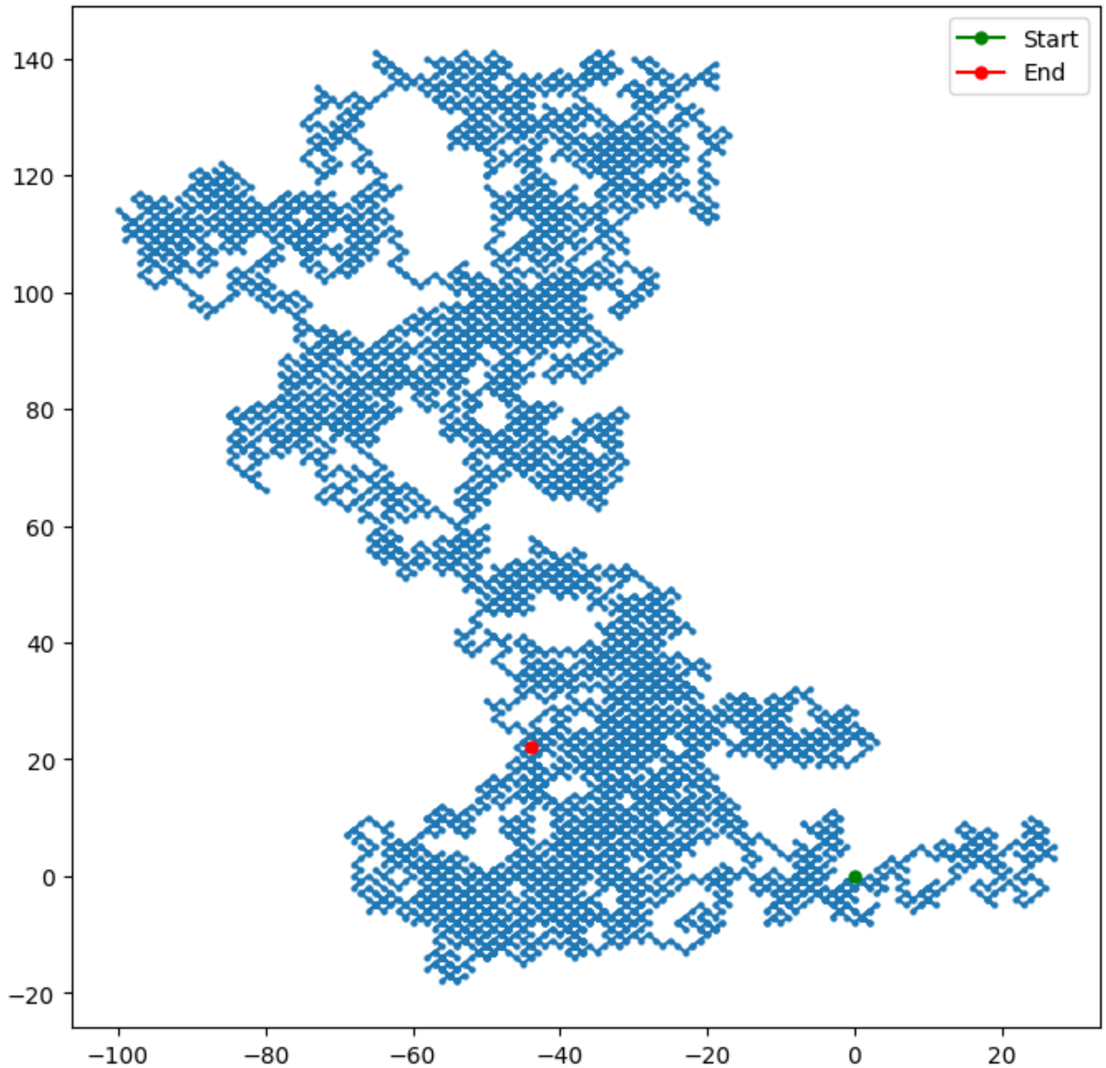
Start
End
40
30
20
10
0
−10
−40
−20
0
20
40



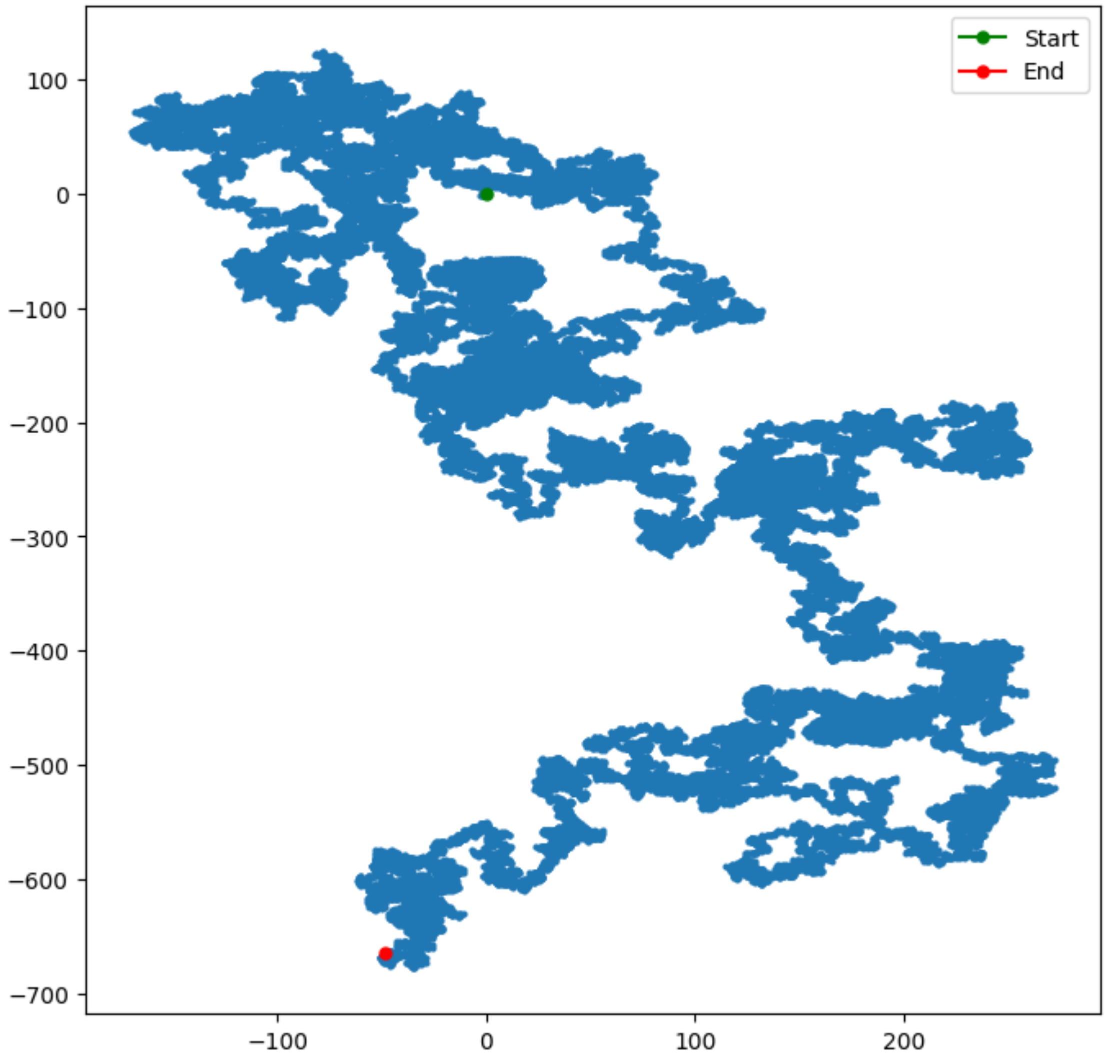
[19]:
```
dxdy = [[1, 1], [-1, -1], [-1, 1], [1, -1]]
for i in n:
    random_walk_2D(i, dxdy)
```


Start
End
10
8
6
4
2
0
−2
−6
−4
−2
0
2


Start
End
60
40
20
0
−20
−50
−40
−30
−20
−10
0

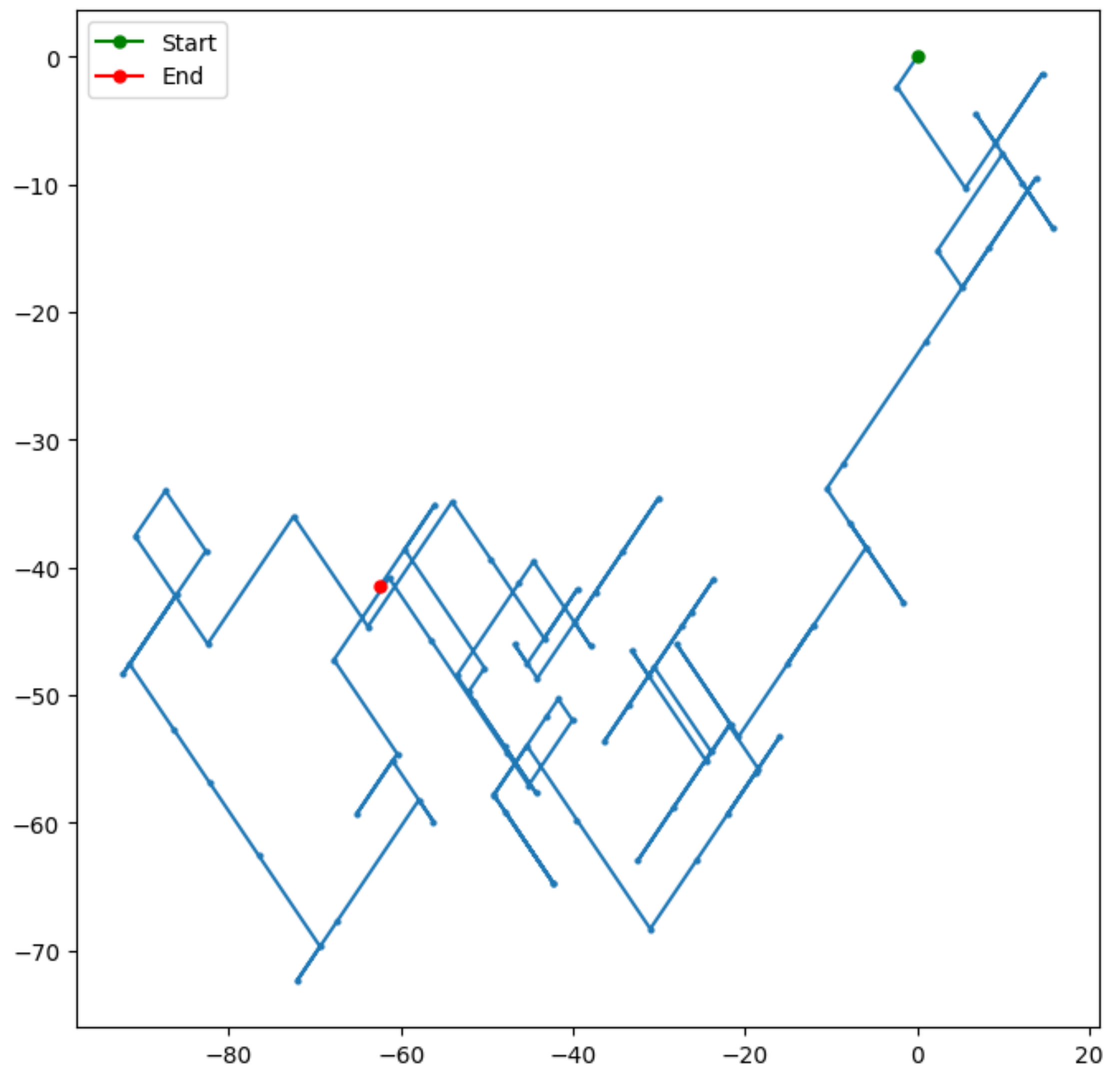


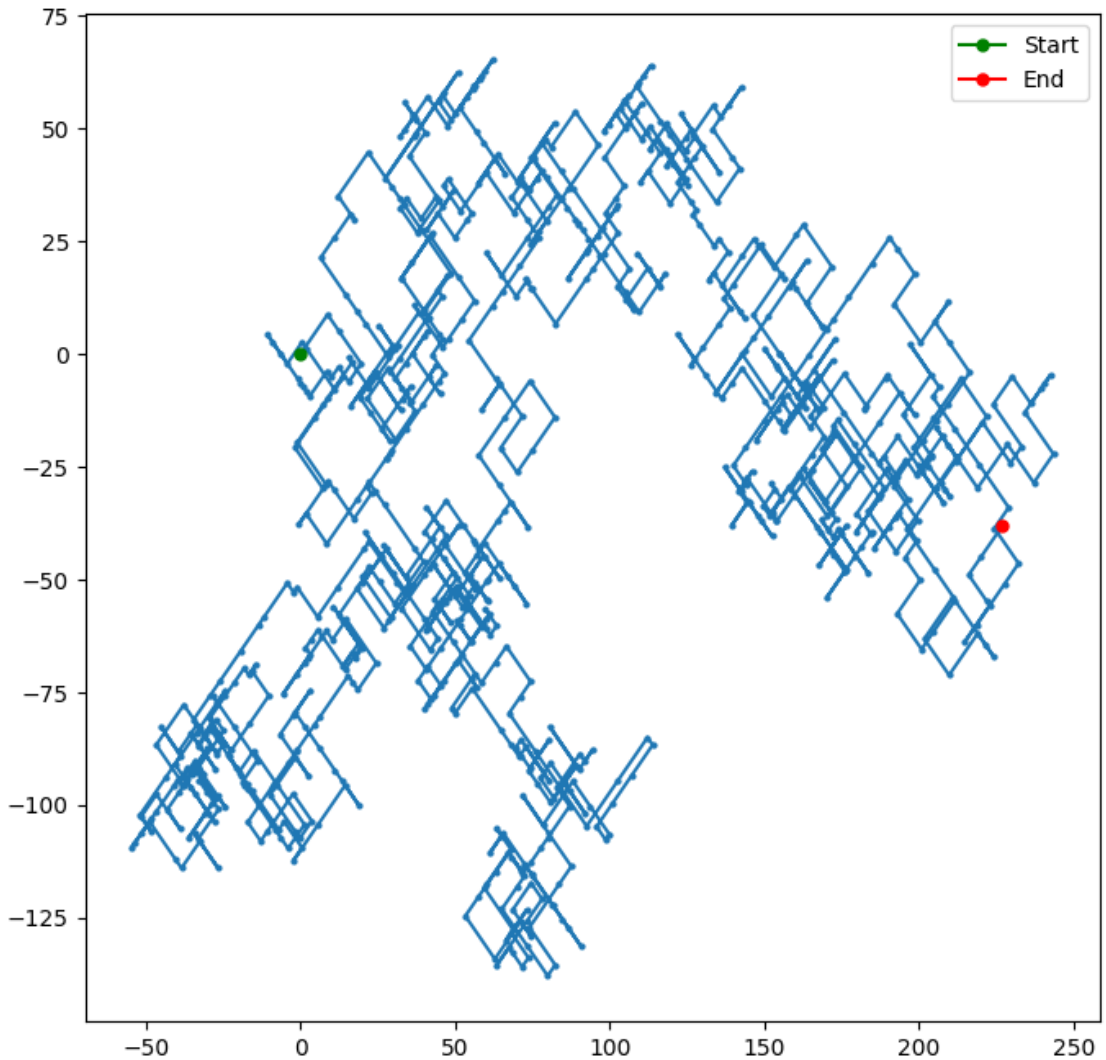
Start
End
140
120
100
80
60
40
20
0
−20
−100
−80
−60
−40
−20
0
20

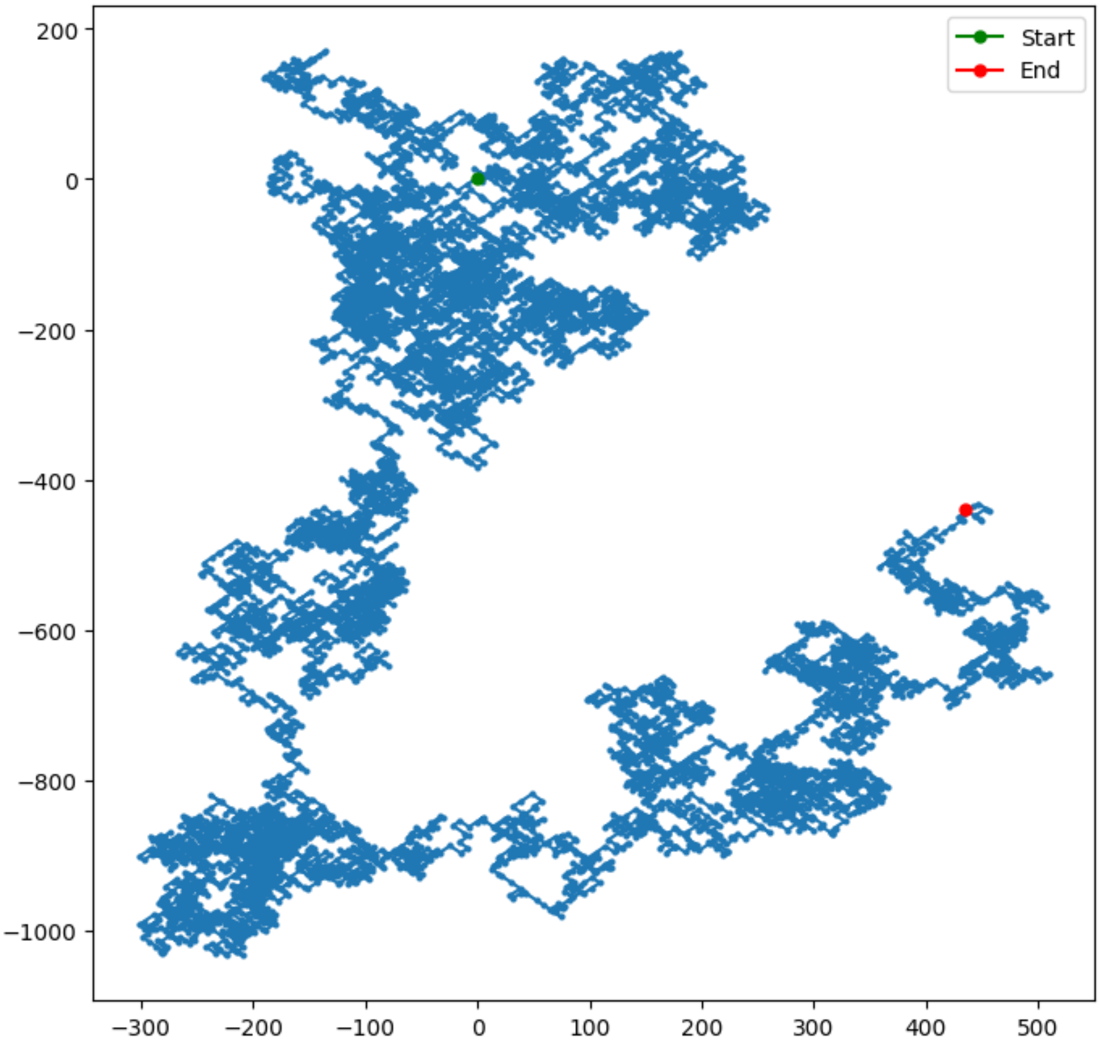



```
[22]: def random_walk_2D_uniform_step(n, dxdy):
    pos = np.zeros((n+1, 2))
    pos[0] = [0, 0]

    for i in range(1, n+1):
        k = np.random.uniform(1, 10)
        x = dxdy[np.random.choice(len(dxdy), p=[1/len(dxdy) for i in␣
↪range(len(dxdy))])]
        pos[i] = pos[i-1] + x * k

    plt.figure(figsize=(8, 8))
    plt.plot(pos[:, 0], pos[:, 1], linestyle='-', marker='o', markersize=2)
    plt.plot(pos[0][0], pos[0][1], marker='o', markersize=5, color='green',␣
↪label='Start')
```

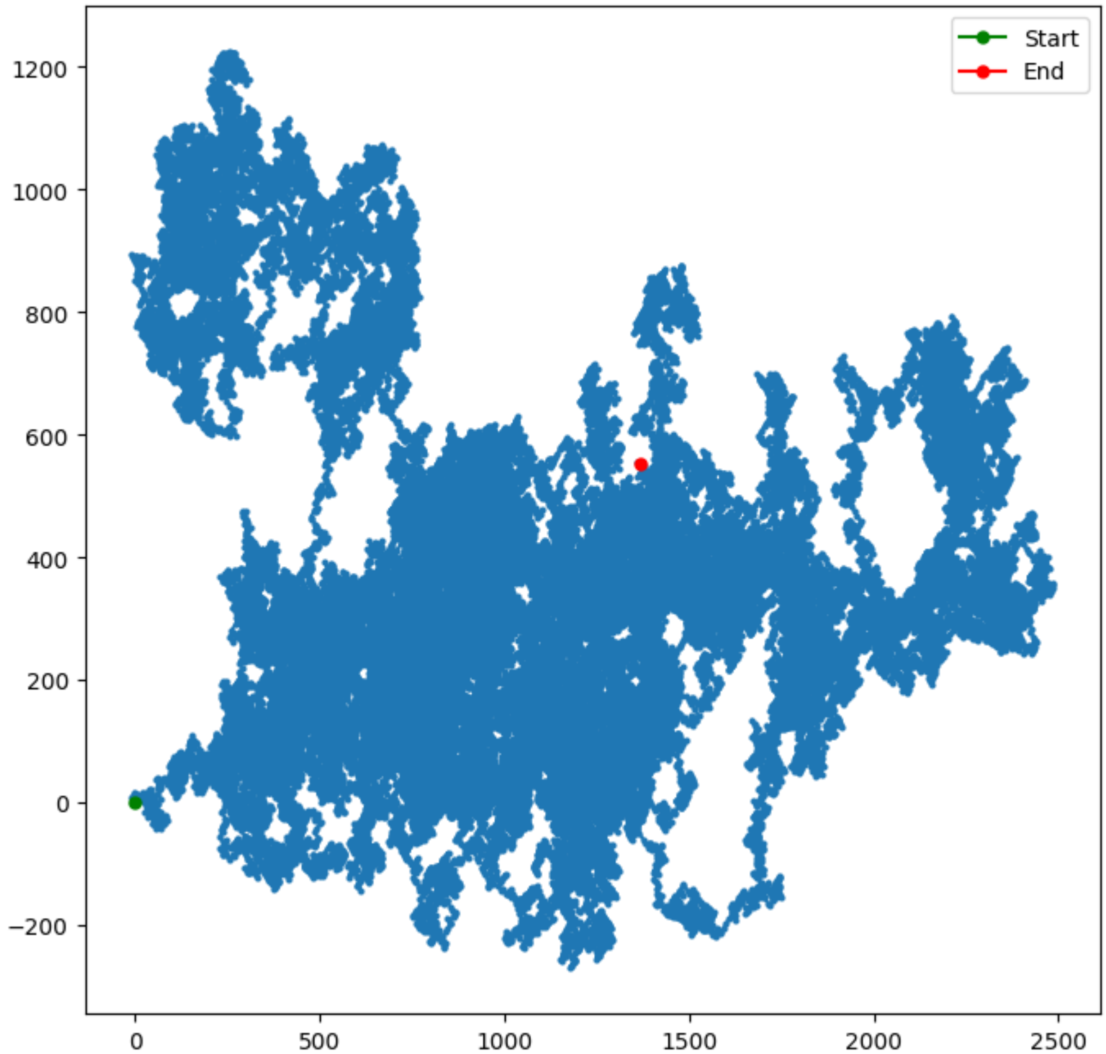
```
    plt.plot(pos[:, 0][-1], pos[:, 1][-1], marker='o', markersize=5,␣
↪color='red', label='End')
    plt.legend()
dxdy = np.array([[1, 1], [-1, -1], [-1, 1], [1, -1]])
for i in n:
    random_walk_2D_uniform_step(i, dxdy)
```




75
50
25
0
−25
−50
−75
−100
−125
−50
0
50
100
150
200
250
Start
End

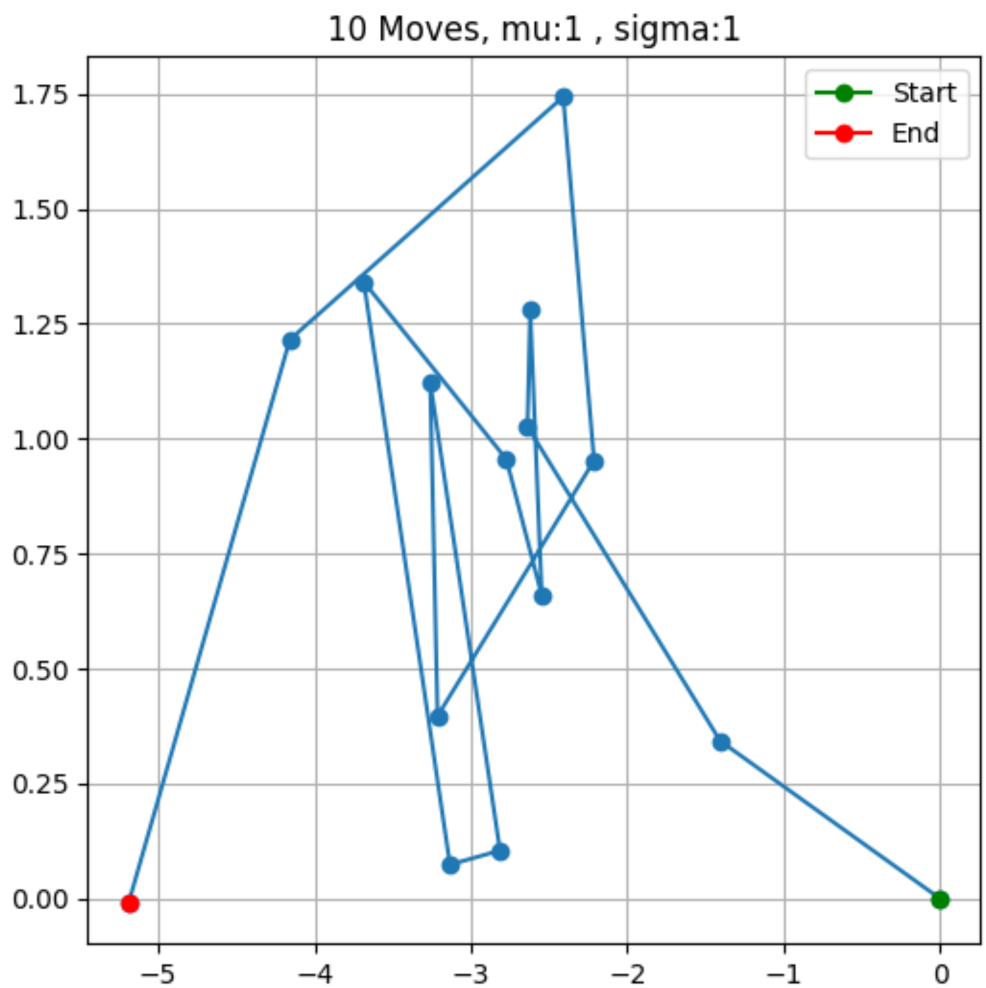


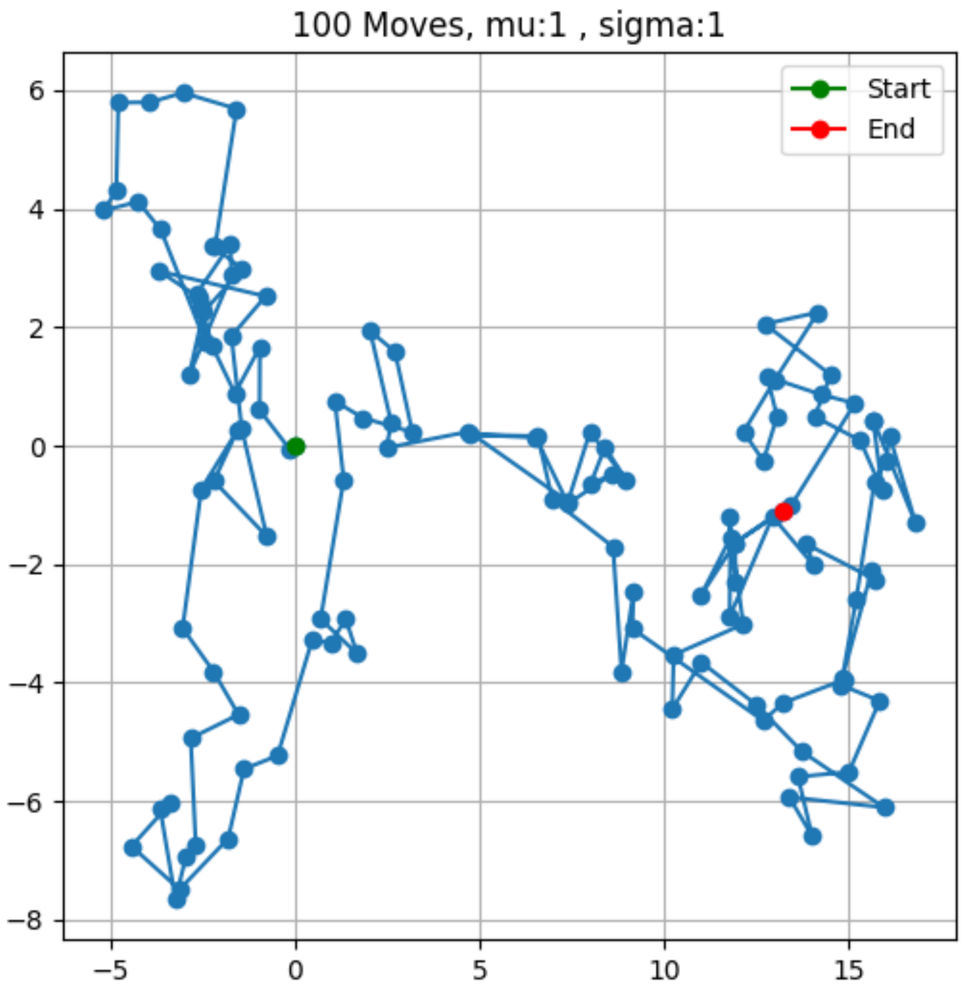
Start
End
200
0
−200
−400
−600
−800
−1000
−300
−200
−100
0
100
200
300
400
500

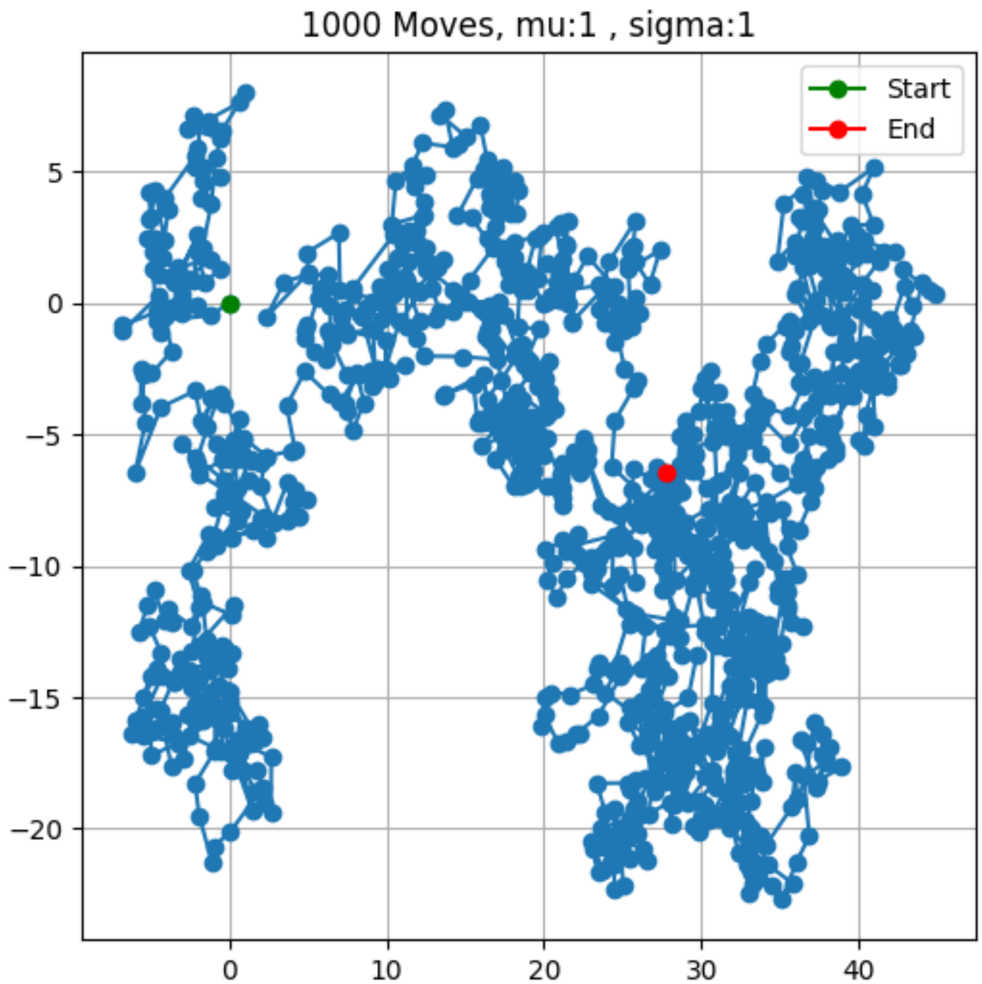


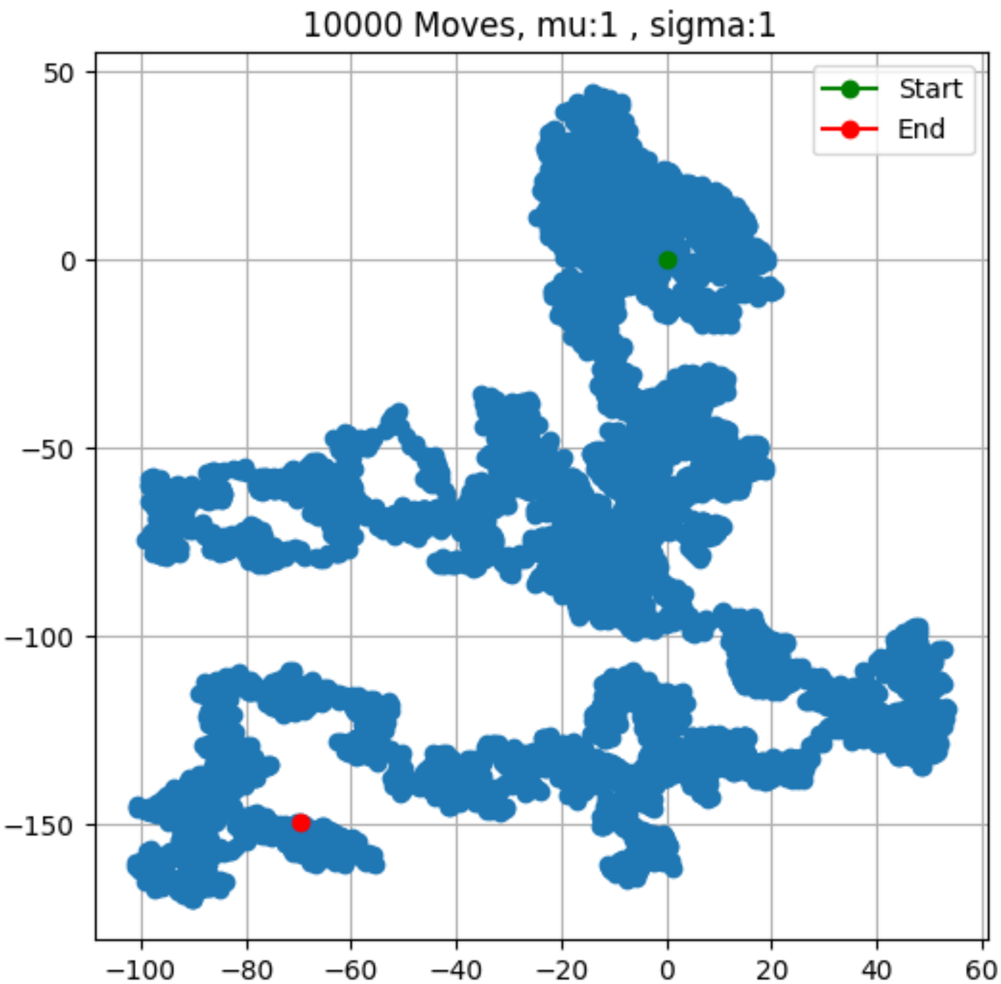
[53]:
```
def simulate_movement(n, mu, sigma):
    positions = [np.array([0, 0])]
    for _ in range(n):
        num_steps = max(1, int(np.random.normal(mu, sigma)))
        move = np.random.normal(0, 1, (num_steps, 2))
        new_positions = positions[-1] + np.cumsum(move, axis=0)
        positions.extend(new_positions.tolist())
    return np.array(positions)
def draw_position_diagram(num_moves, mu, sigma):
    positions = simulate_movement(num_moves, mu, sigma)
    plt.figure(figsize=(6,6))
    plt.plot(positions[:,0], positions[:,1], marker='o')
    plt.plot(positions[0,0], positions[0,1], marker='o', color='green',␣
 ↪label='Start')
```

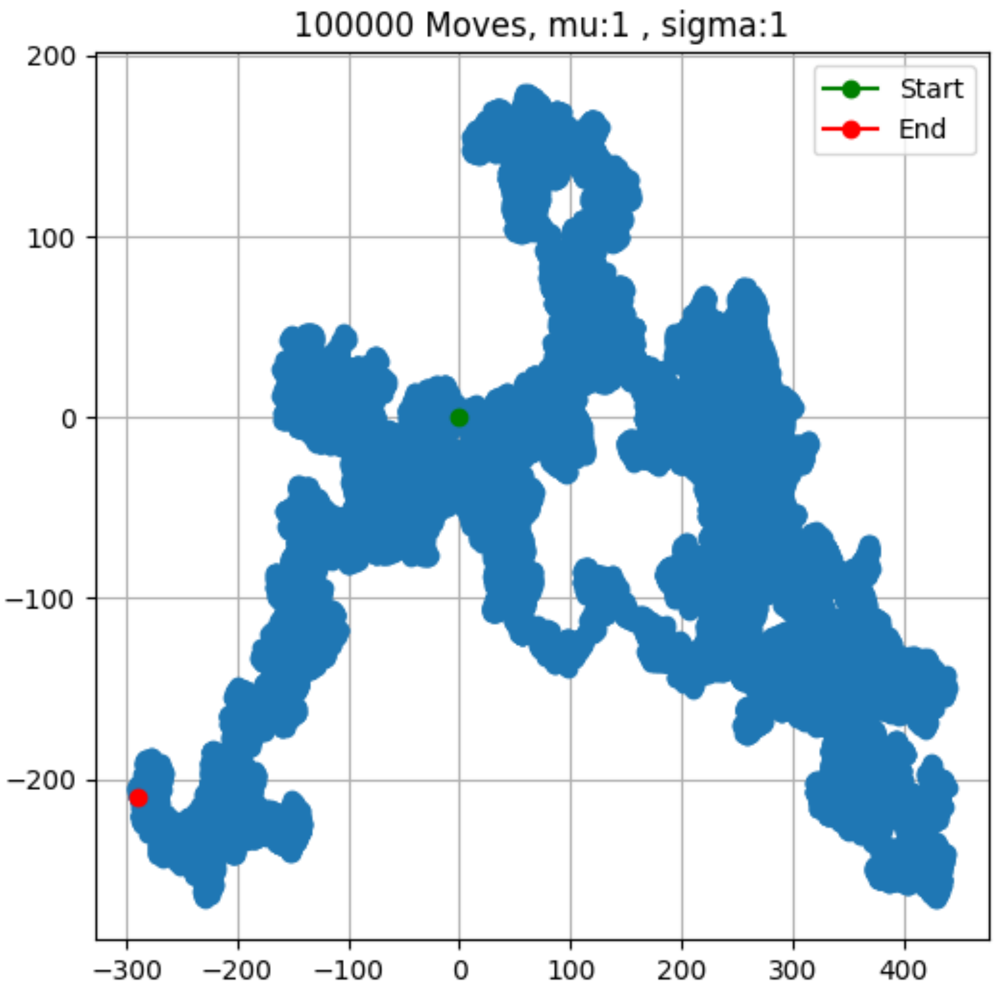
```
    plt.plot(positions[-1,0], positions[-1,1], marker='o', color='red',␣
↪label='End')
    plt.title(f'{num_moves} Moves, mu:{mu} , sigma:{sigma}')
    plt.legend()
    plt.grid(True)
    plt.show()
mu_values = [1,-5]
sigma_values = [1, 2 , 5, 10,20,25,50,100]
for mu in mu_values:
    for sigma in sigma_values:
        draw_position_diagram(10, mu, sigma)
        draw_position_diagram(100, mu, sigma)
        draw_position_diagram(1000, mu, sigma)
        draw_position_diagram(10000, mu, sigma)
        draw_position_diagram(100000, mu, sigma)
```

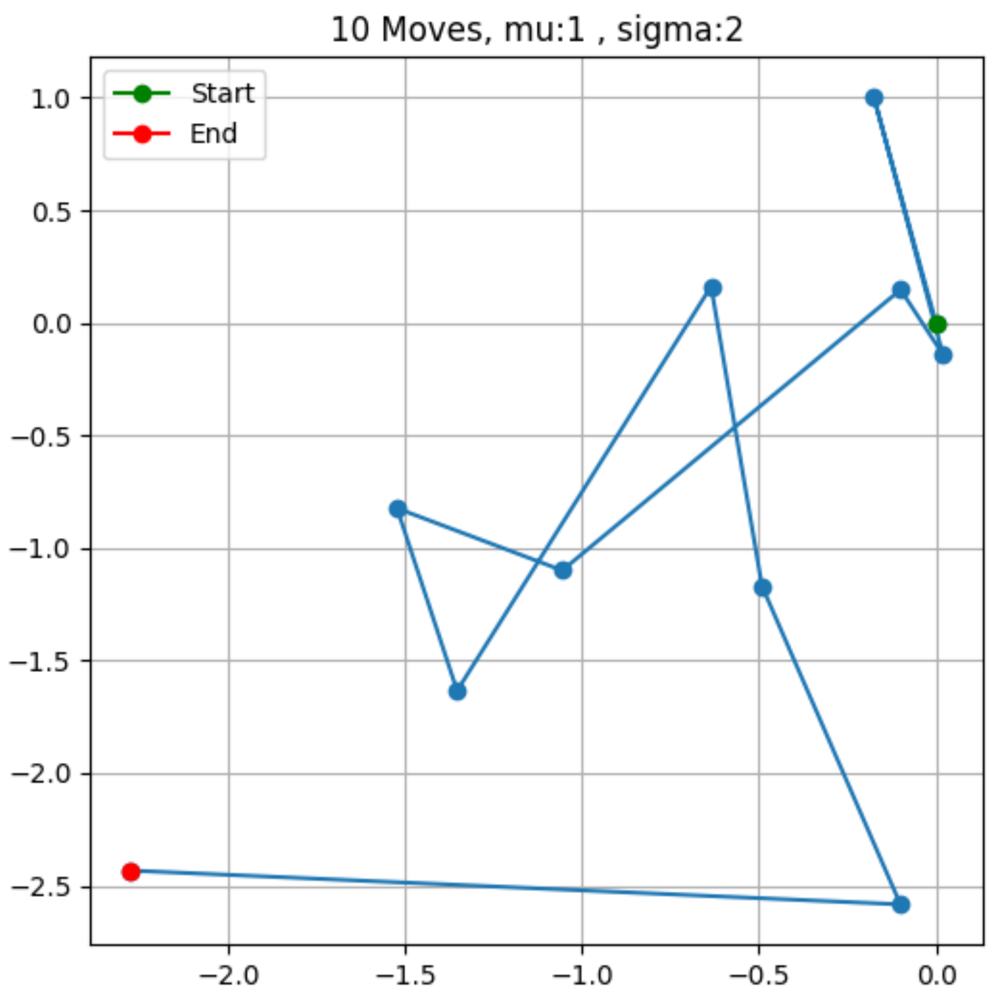


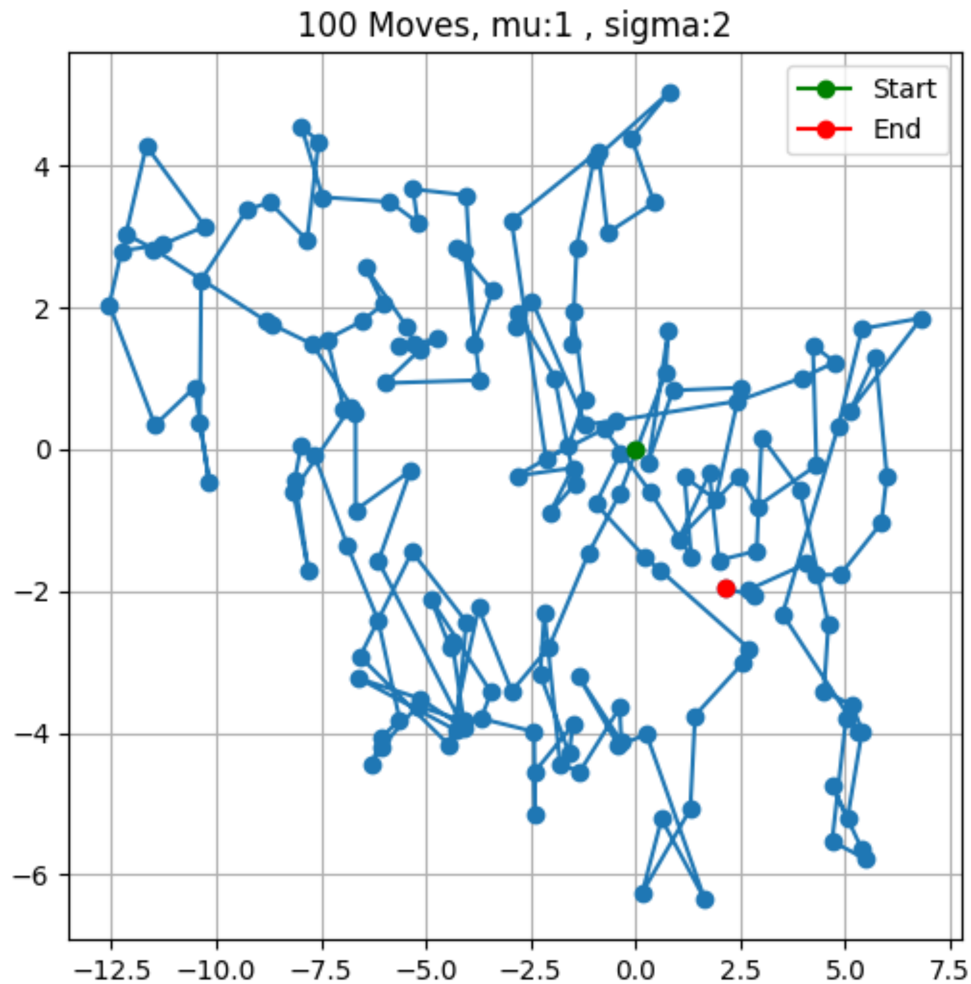

100 Moves, mu:1 , sigma:1
Start
End
6
4
2
0
−2
−4
−6
−8
−5
0
5
10
15

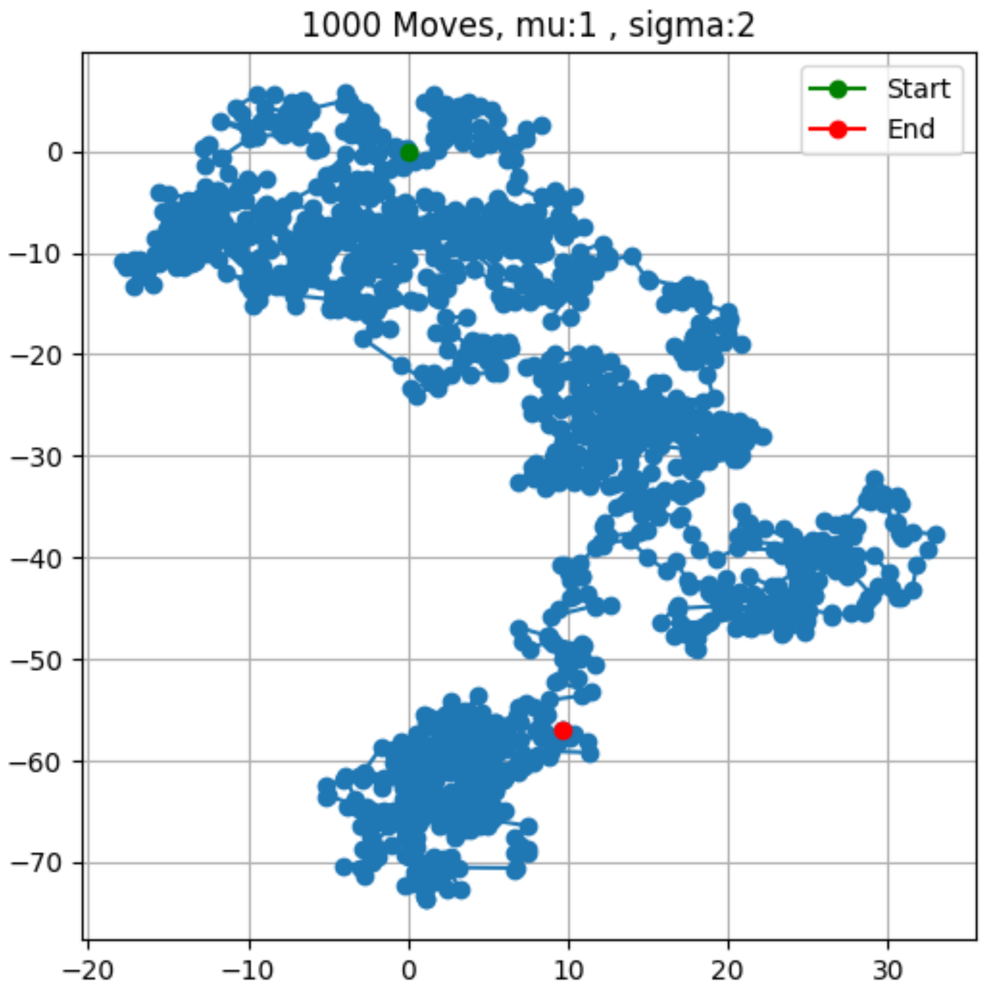

1000 Moves, mu:1 , sigma:1
Start
End
5
0
−5
−10
−15
−20
0
10
20
30
40

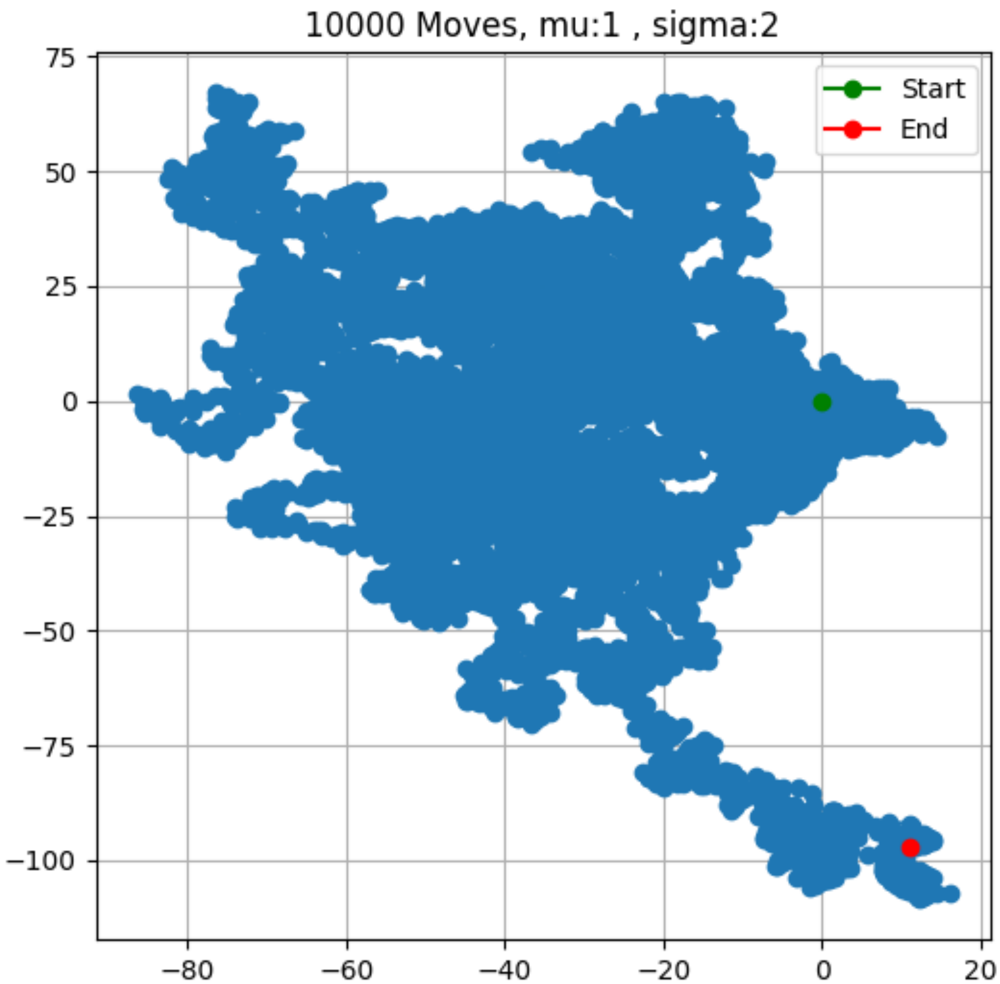

10000 Moves, mu:1 , sigma:1
Start
End
50
0
−50
−100
−150
−100
−80
−60
−40
−20
0
20
40
60

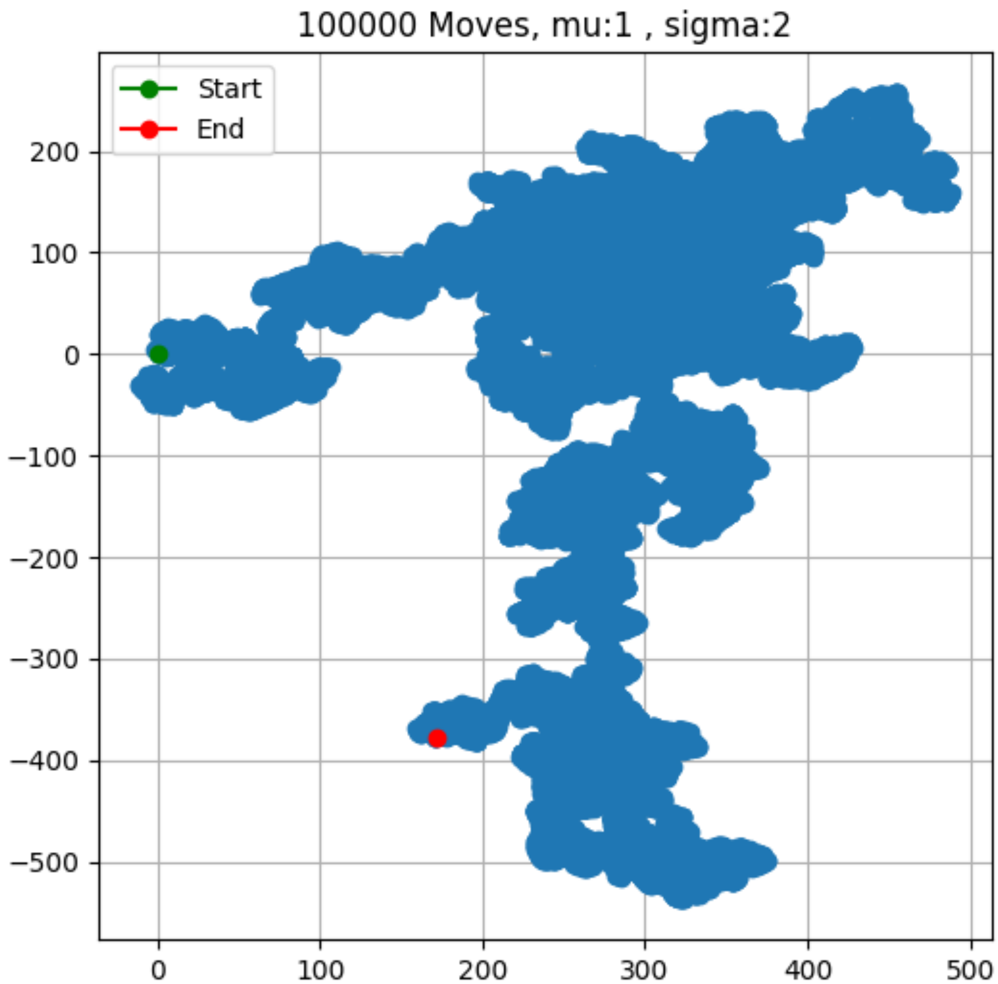

100000 Moves, mu:1 , sigma:1
Start
End
200
100
0
−100
−200
−300
−200
−100
0
100
200
300
400

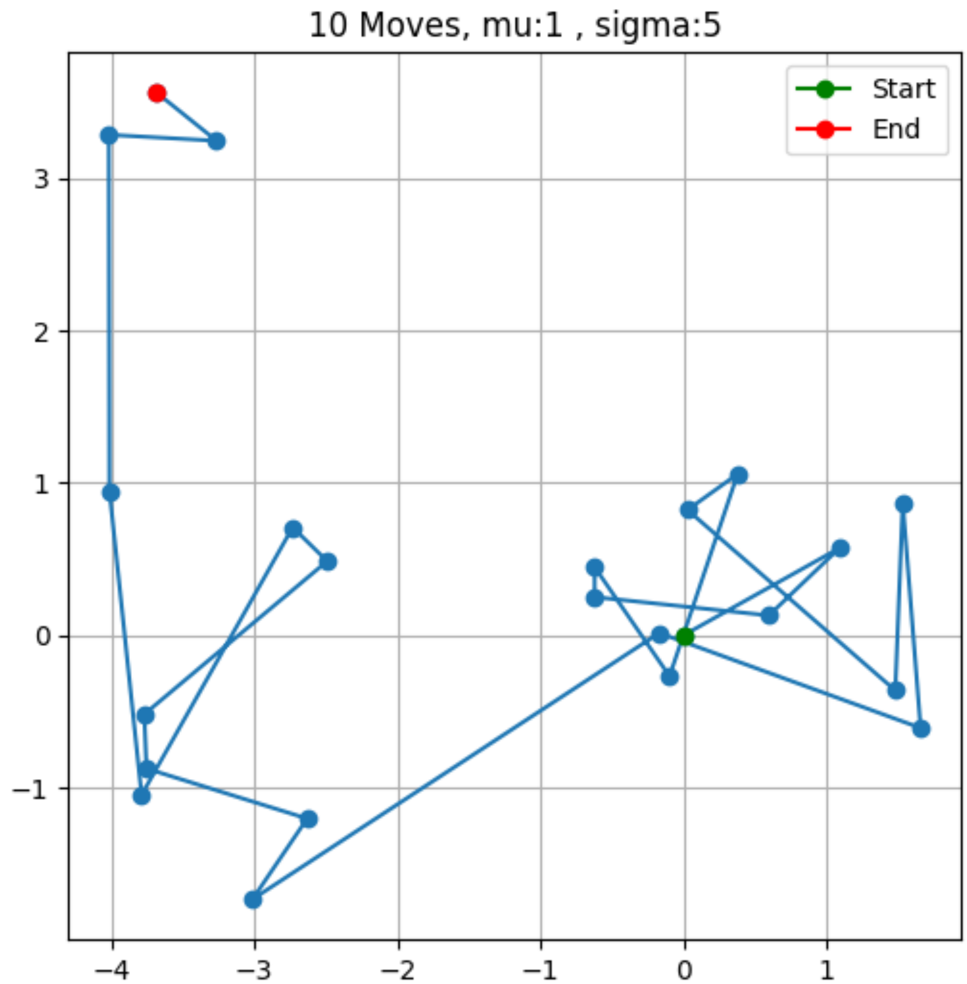

10 Moves, mu:1 , sigma:2
Start
End
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−2.0
−1.5
−1.0
−0.5
0.0

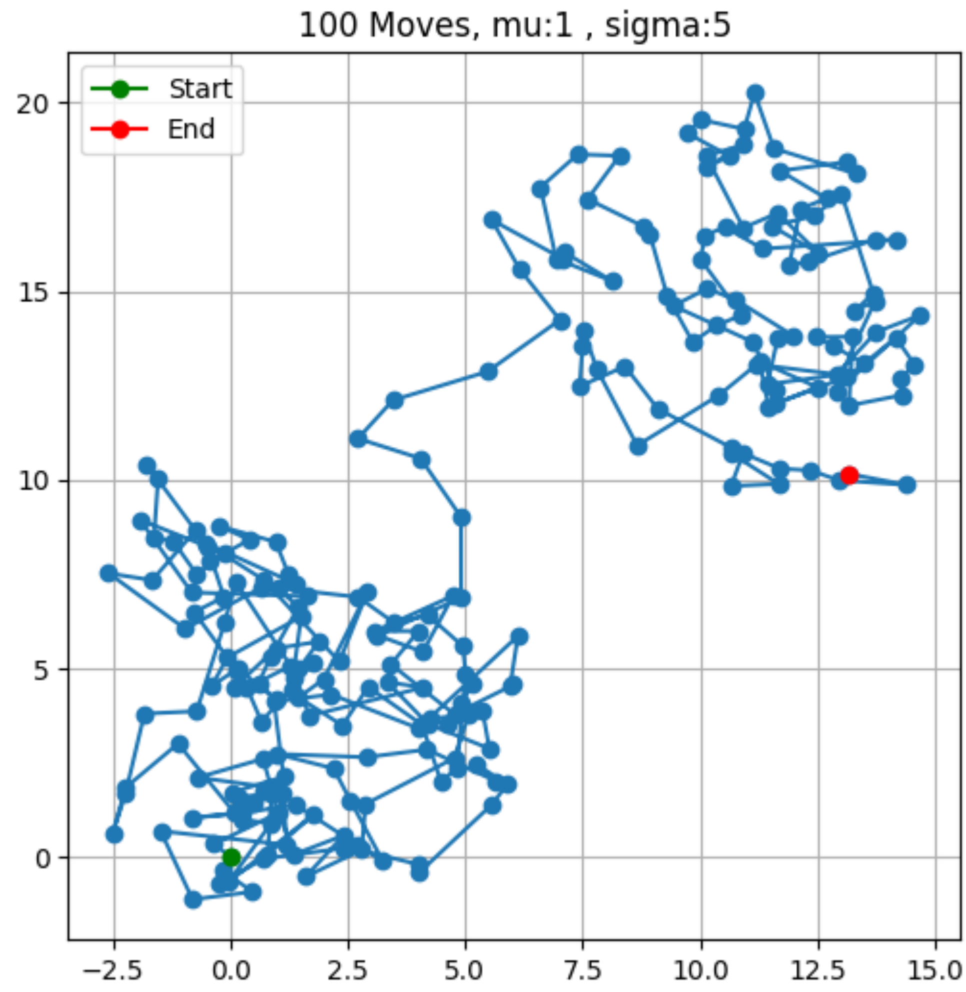

100 Moves, mu:1 , sigma:2
Start
End
4
2
0
−2
−4
−6
−12.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5

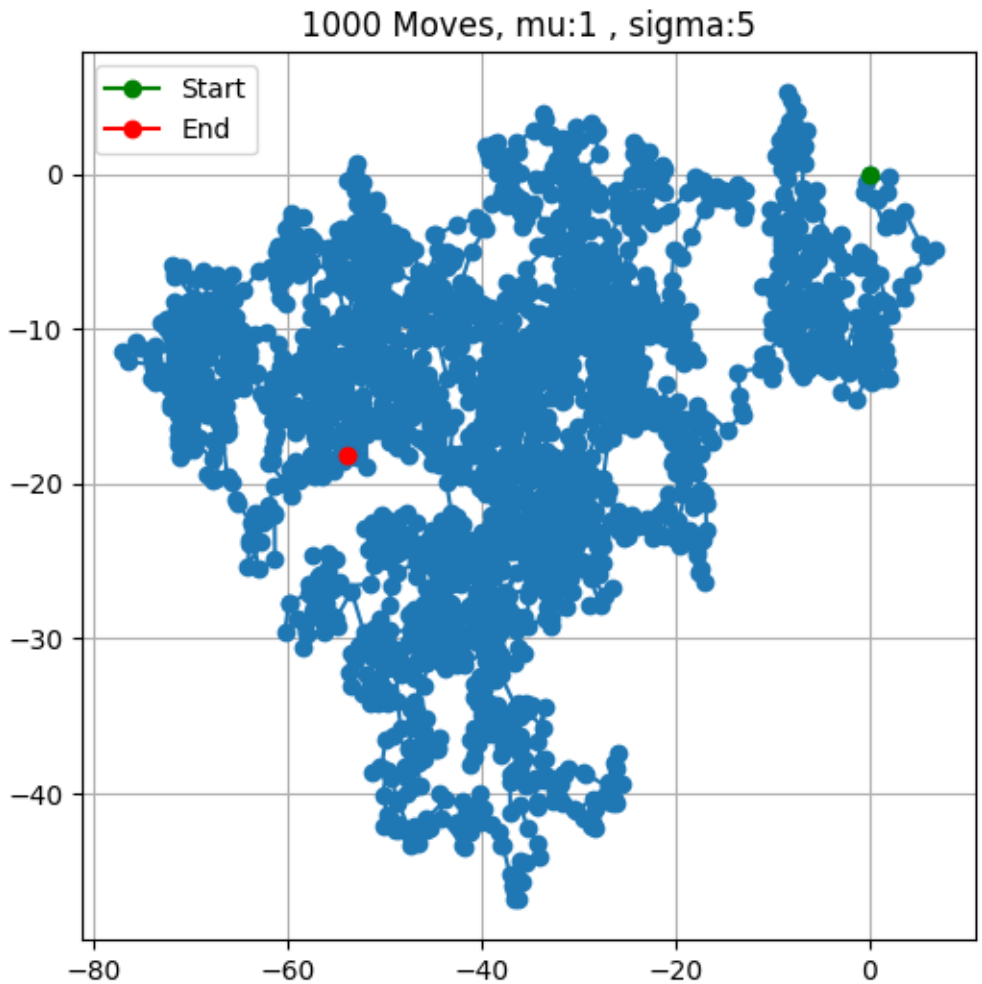

1000 Moves, mu:1 , sigma:2
Start
End
0
−10
−20
−30
−40
−50
−60
−70
−20
−10
0
10
20
30

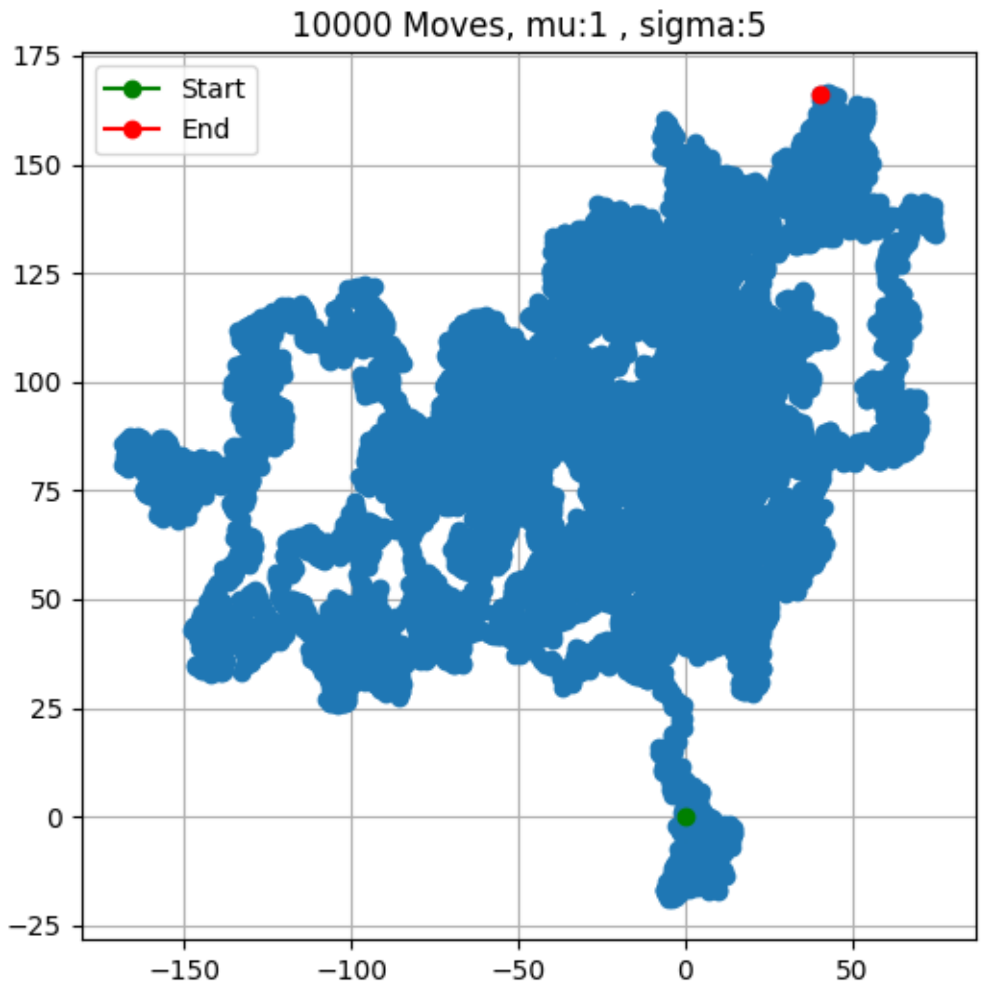

10000 Moves, mu:1 , sigma:2
Start
End
75
50
25
0
−25
−50
−75
−100
−80
−60
−40
−20
0
20

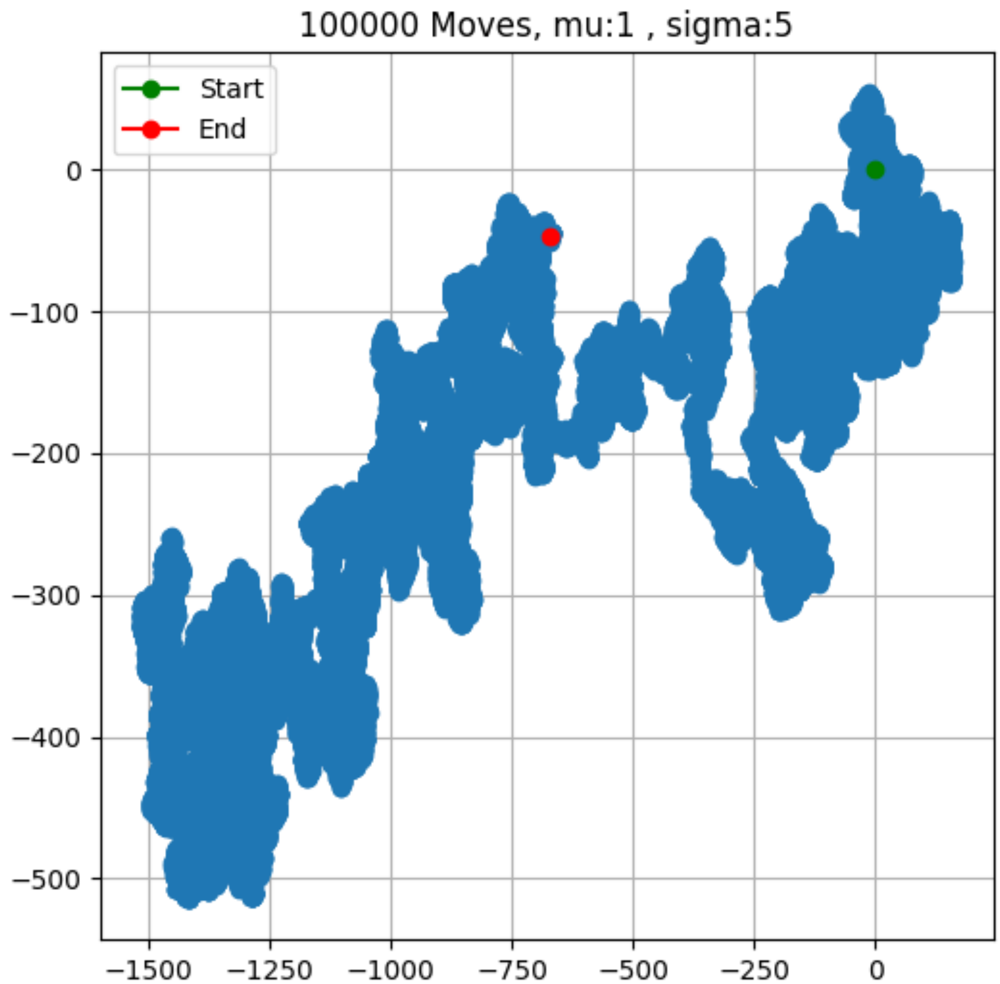

100000 Moves, mu:1 , sigma:2
Start
End
200
100
0
−100
−200
−300
−400
−500
0
100
200
300
400
500

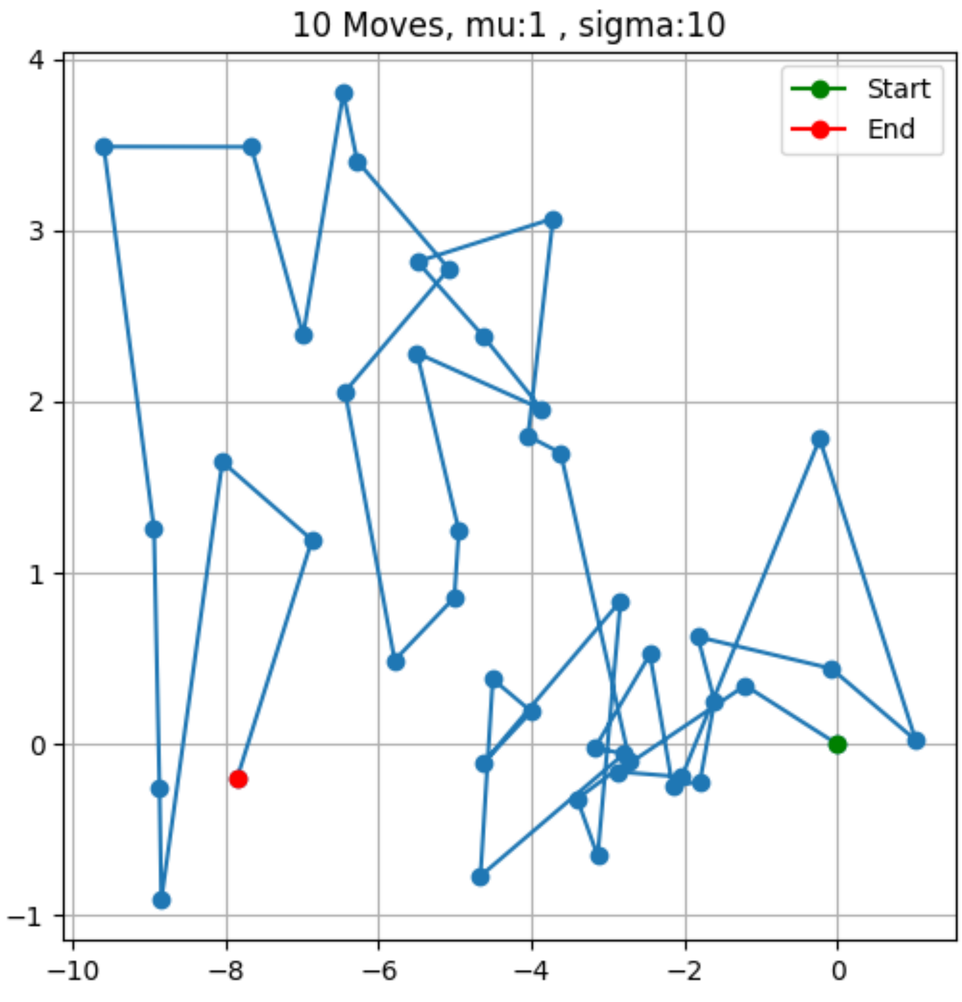

10 Moves, mu:1 , sigma:5
Start
End
3
2
1
0
−1
−4
−3
−2
−1
0
1

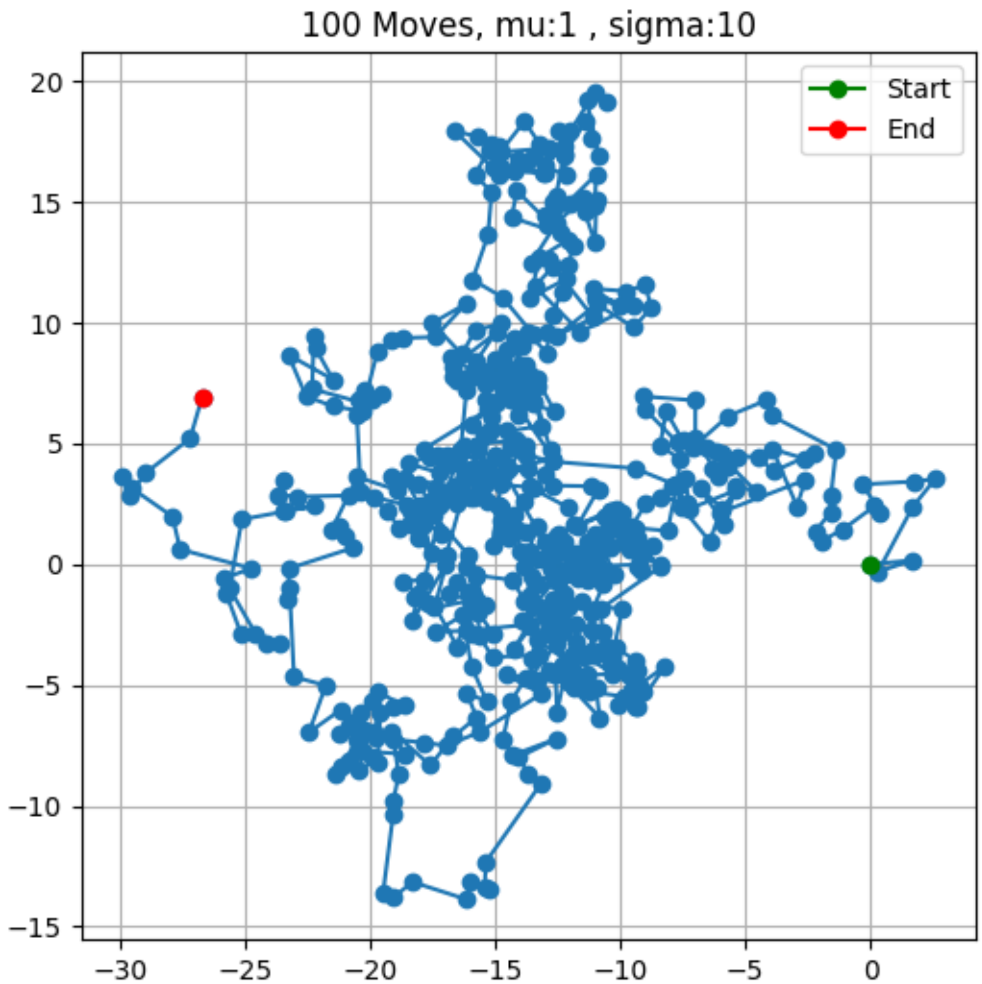

100 Moves, mu:1 , sigma:5
Start
End
20
15
10
5
0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0


1000 Moves, mu:1 , sigma:5
Start
End
0
−10
−20
−30
−40
−80
−60
−40
−20
0

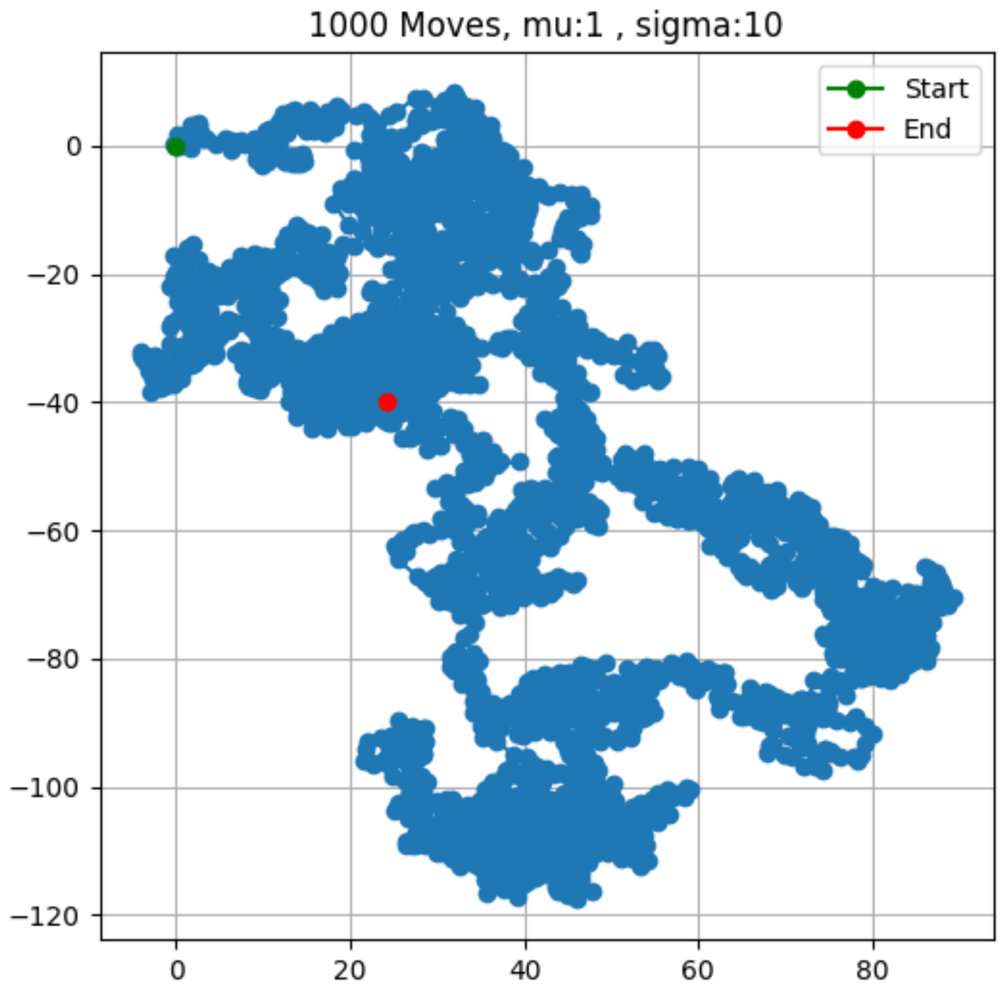

10000 Moves, mu:1 , sigma:5
Start
End
175
150
125
100
75
50
25
0
−25
−150
−100
−50
0
50

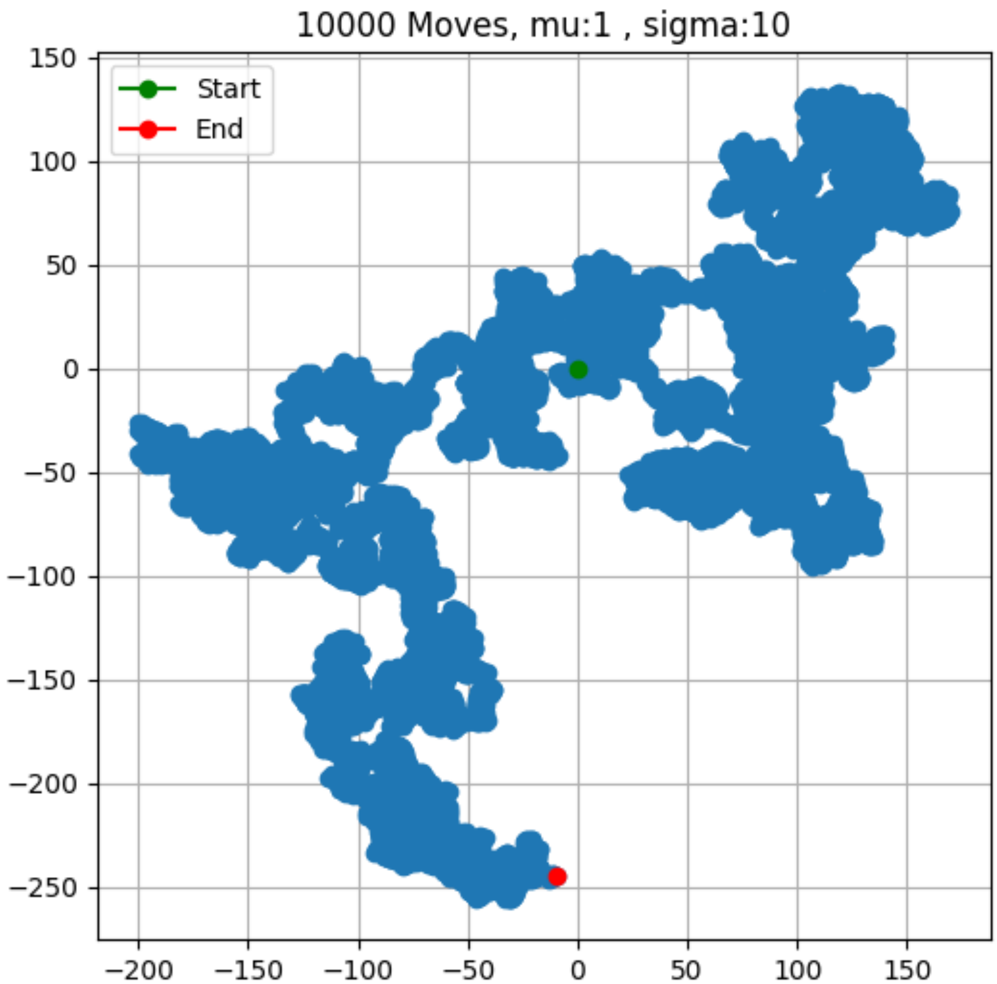

100000 Moves, mu:1 , sigma:5
Start
End
0
−100
−200
−300
−400
−500
−1500
−1250
−1000
−750
−500
−250
0


10 Moves, mu:1 , sigma:10
Start
End
4
3
2
1
0
−1
−10
−8
−6
−4
−2
0

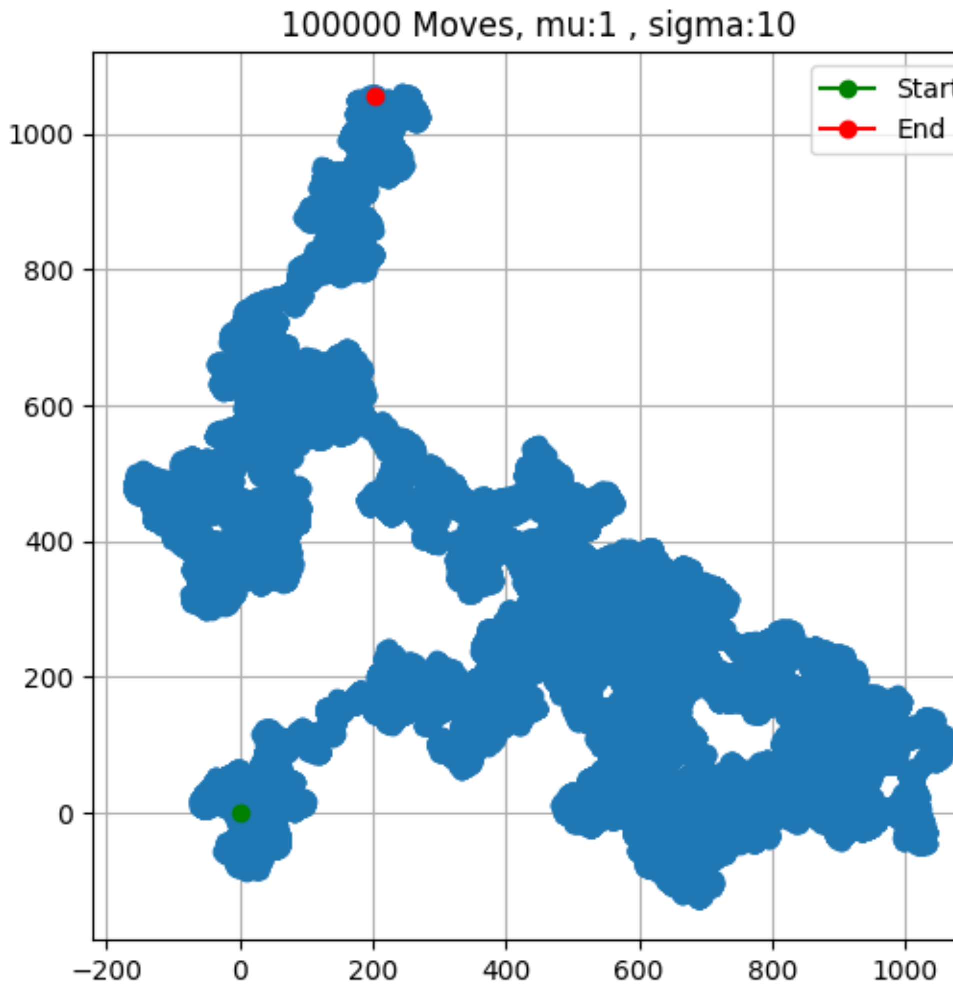


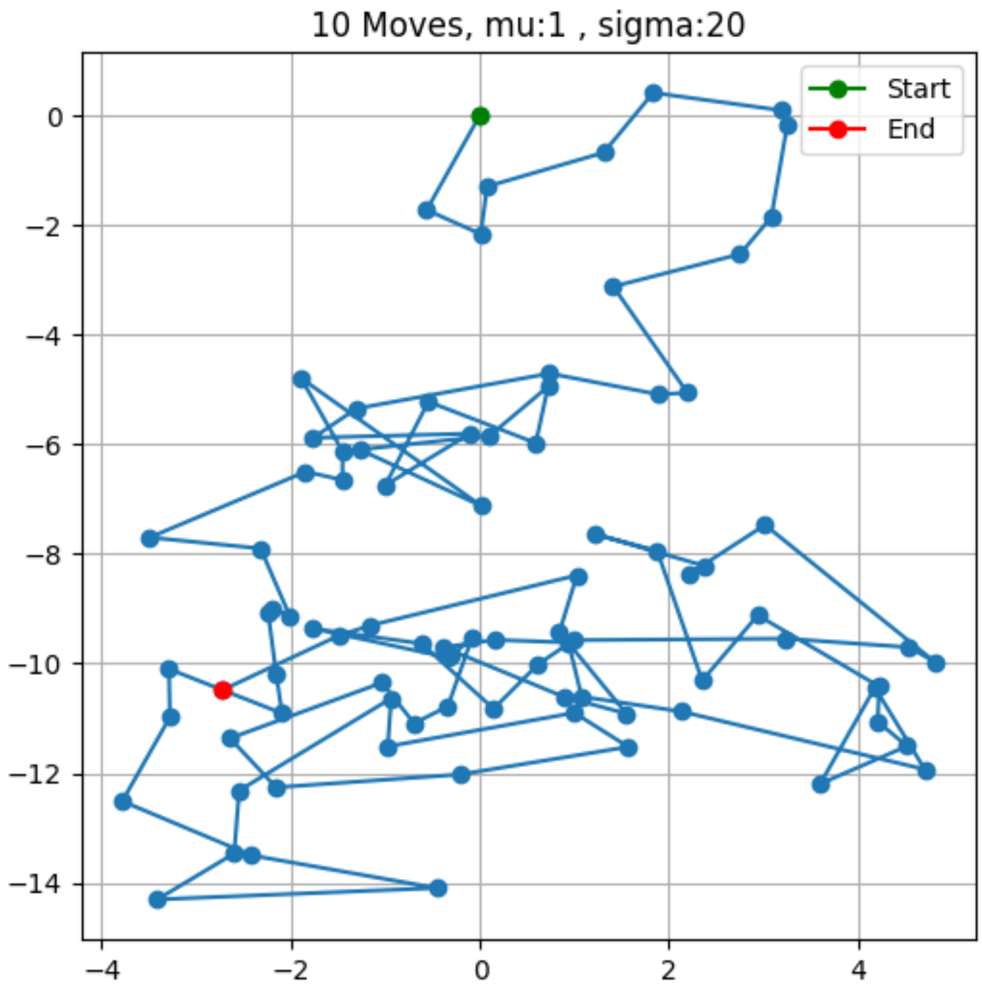
100 Moves, mu:1 , sigma:10
Start
End
20
15
10
5
0
−5
−10
−15
−30
−25
−20
−15
−10
−5
0

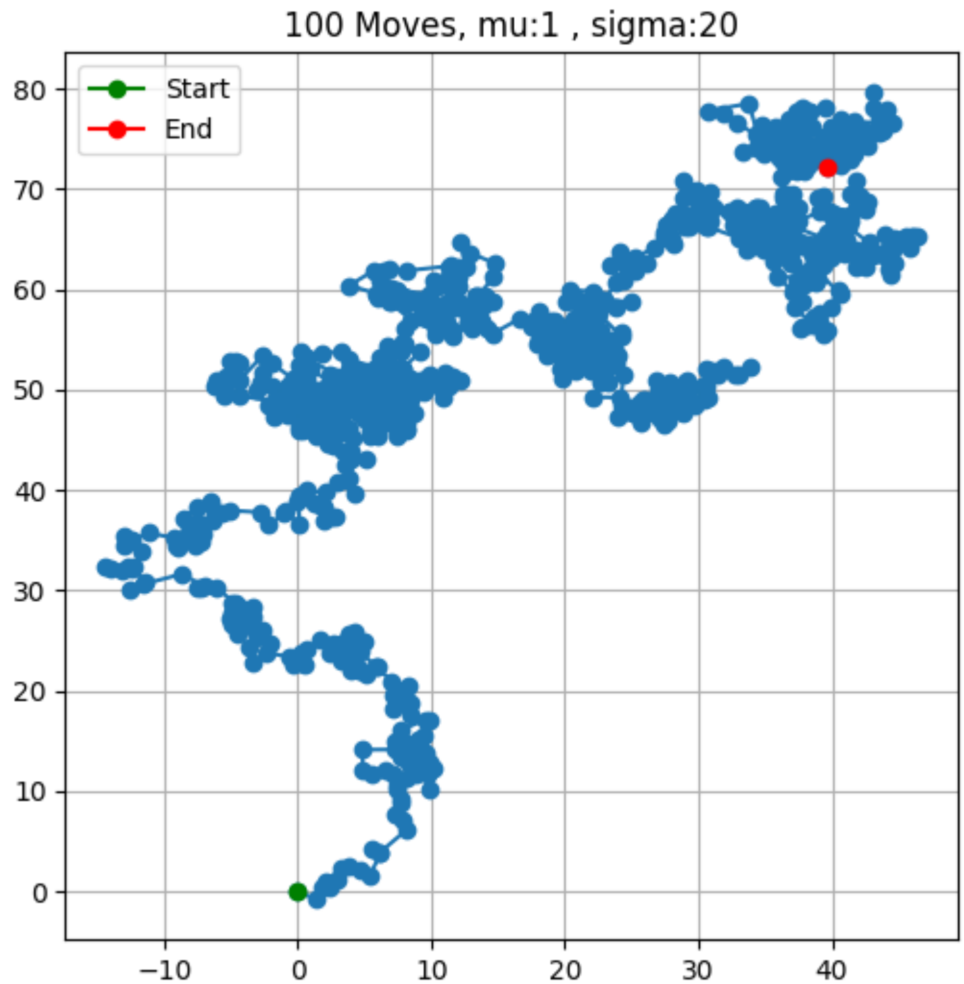

1000 Moves, mu:1 , sigma:10
Start
End
0
−20
−40
−60
−80
−100
−120
0
20
40
60
80

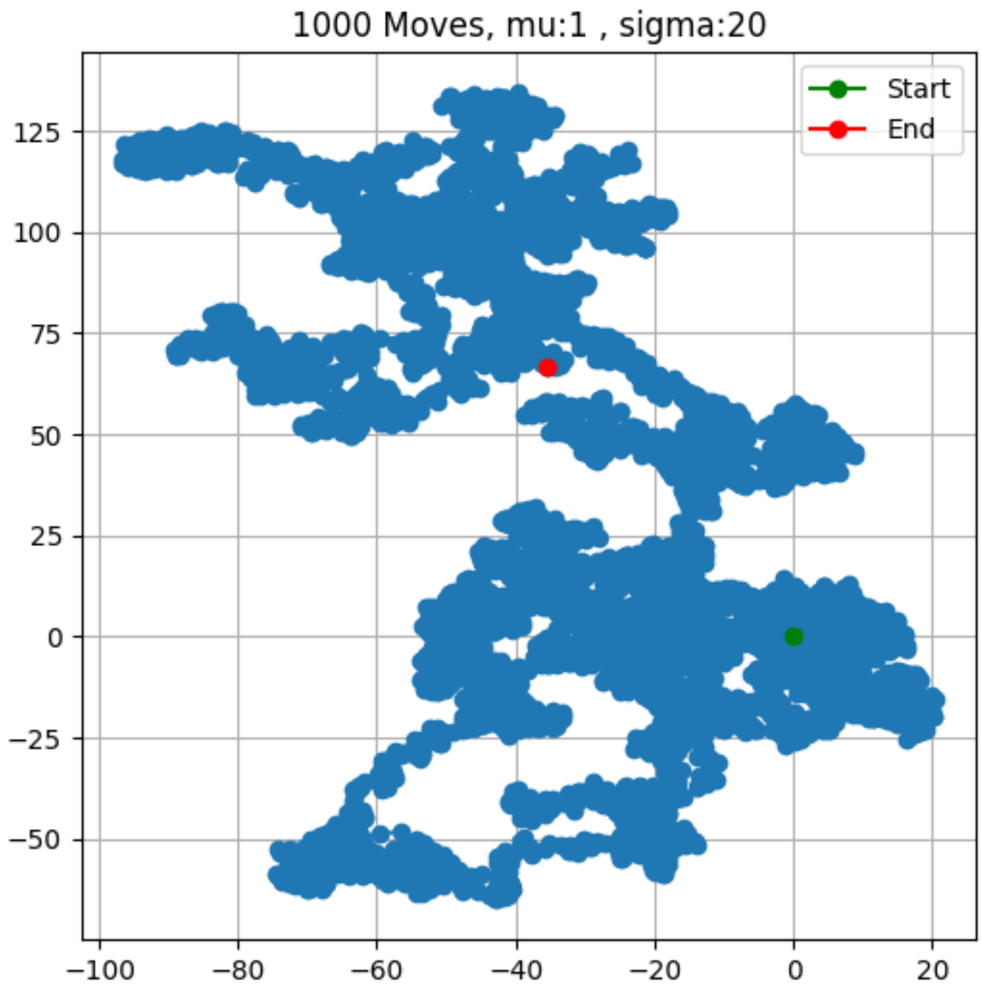


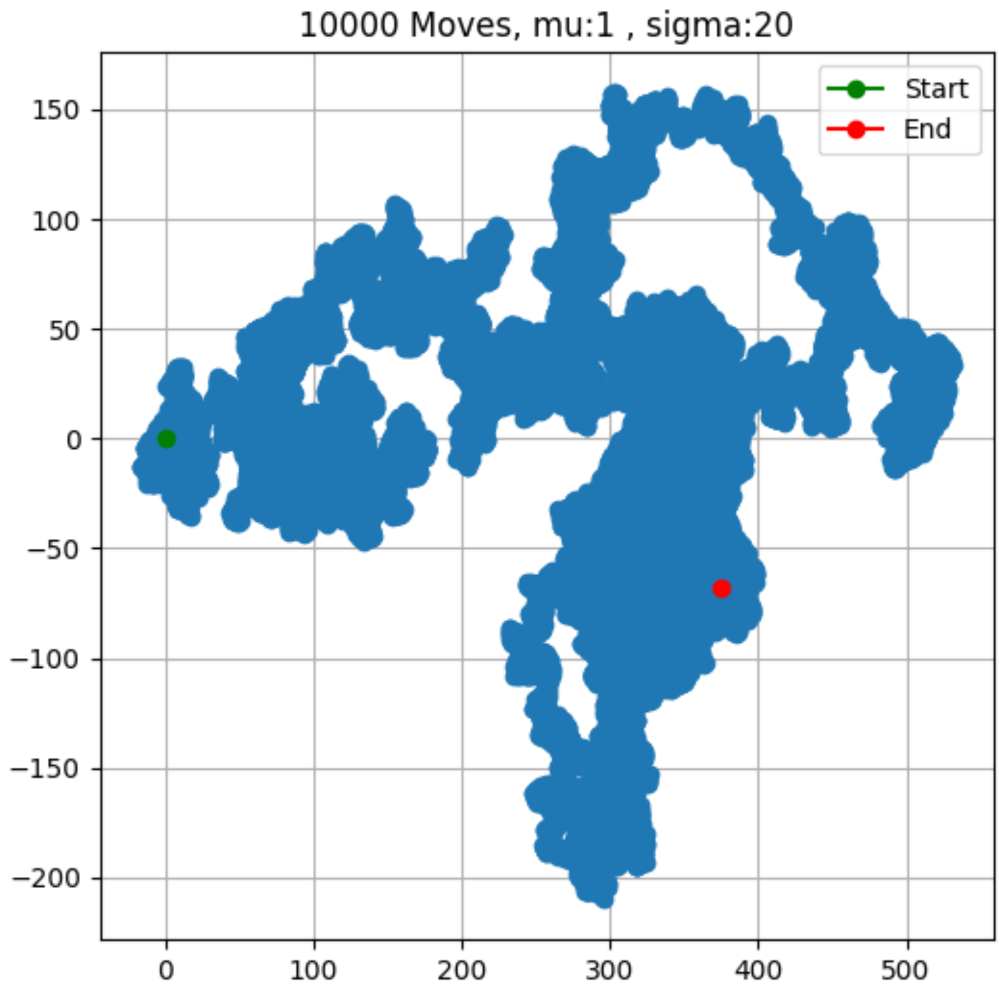
10000 Moves, mu:1 , sigma:10
Start
End
150
100
50
0
−50
−100
−150
−200
−250
−200
−150
−100
−50
0
50
100
150

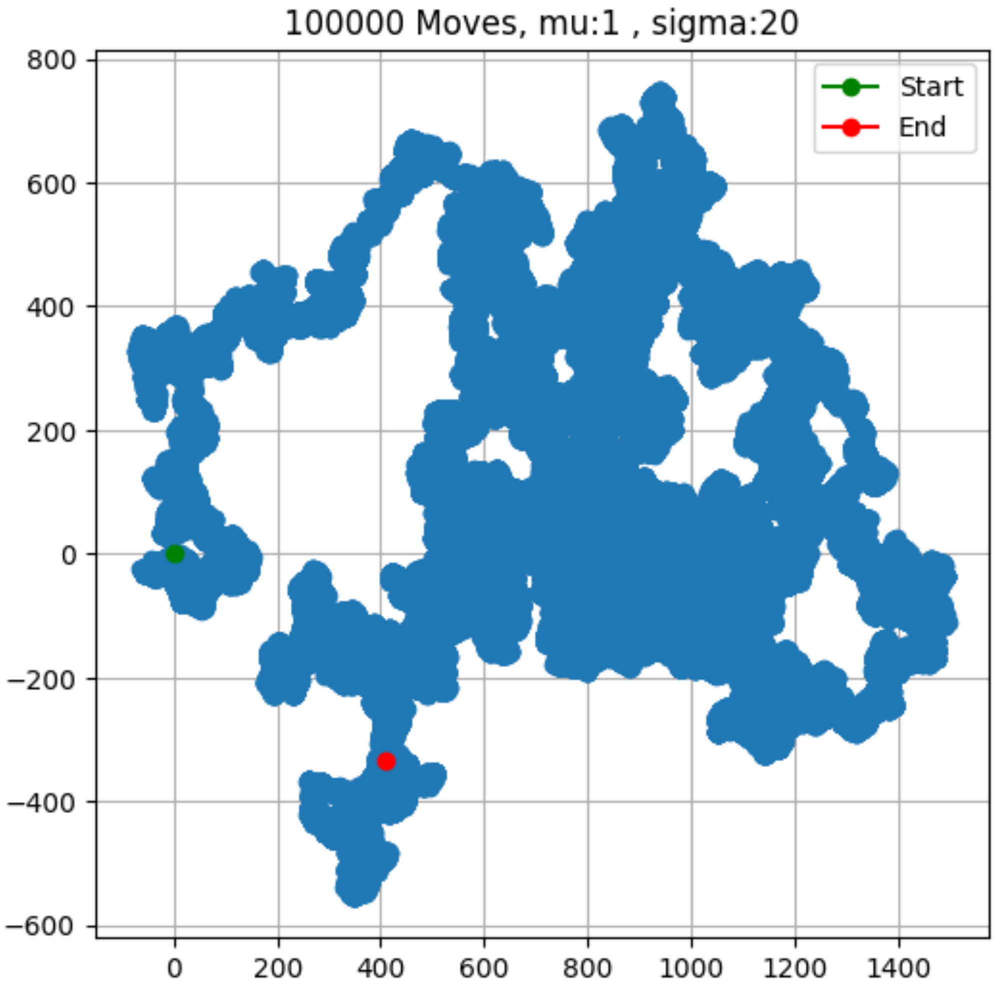

100000 Moves, mu:1 , sigma:10
Start
End
1000
800
600
400
200
0
−200
0
200
400
600
800
1000


10 Moves, mu:1 , sigma:20
Start
End
0
−2
−4
−6
−8
−10
−12
−14
−4
−2
0
2
4

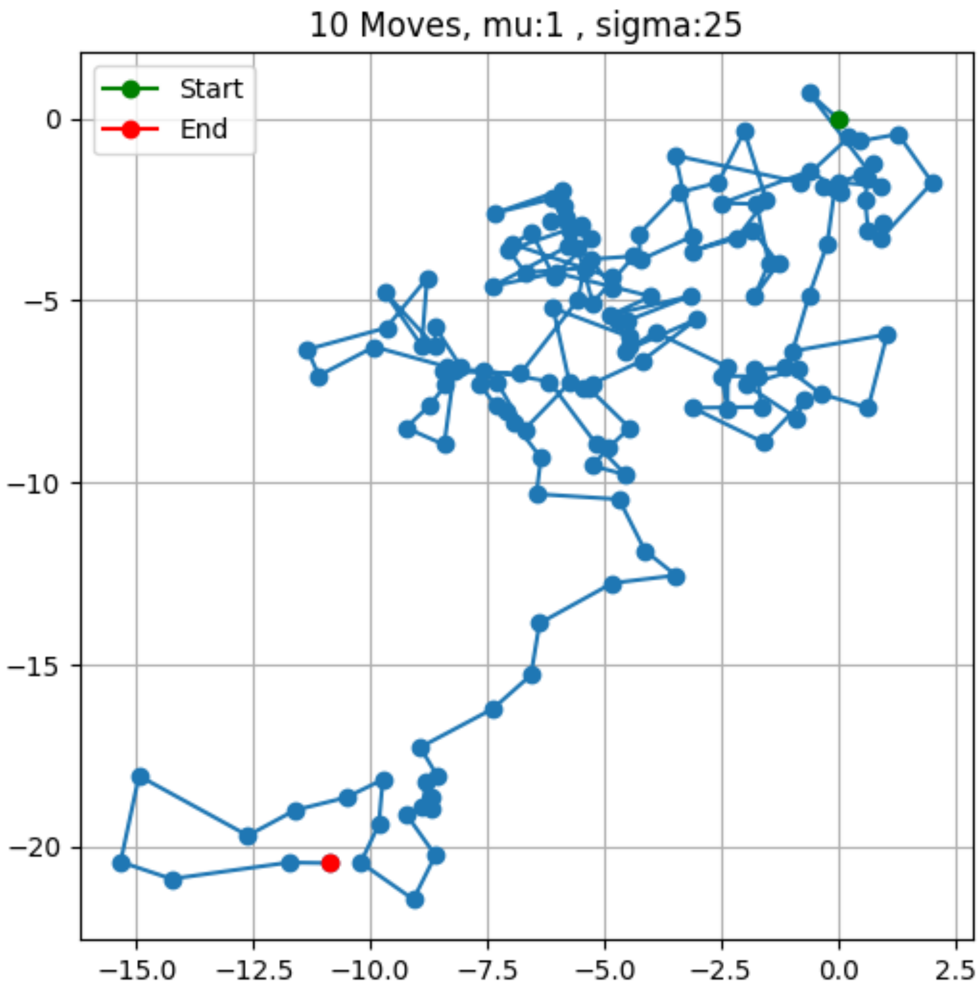

100 Moves, mu:1 , sigma:20
Start
End
80
70
60
50
40
30
20
10
0
−10
0
10
20
30
40


1000 Moves, mu:1 , sigma:20
Start
End
125
100
75
50
25
0
−25
−50
−100
−80
−60
−40
−20
0
20

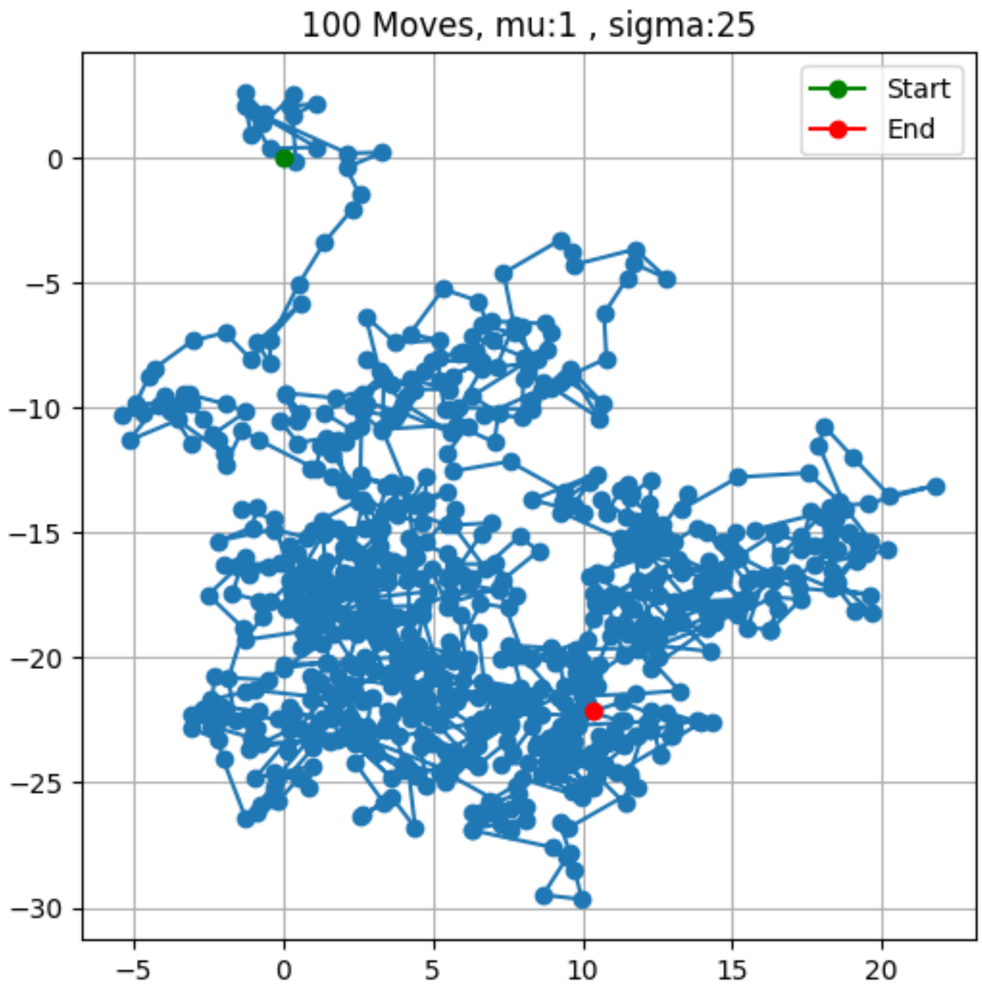

10000 Moves, mu:1 , sigma:20
Start
End
150
100
50
0
−50
−100
−150
−200
0
100
200
300
400
500

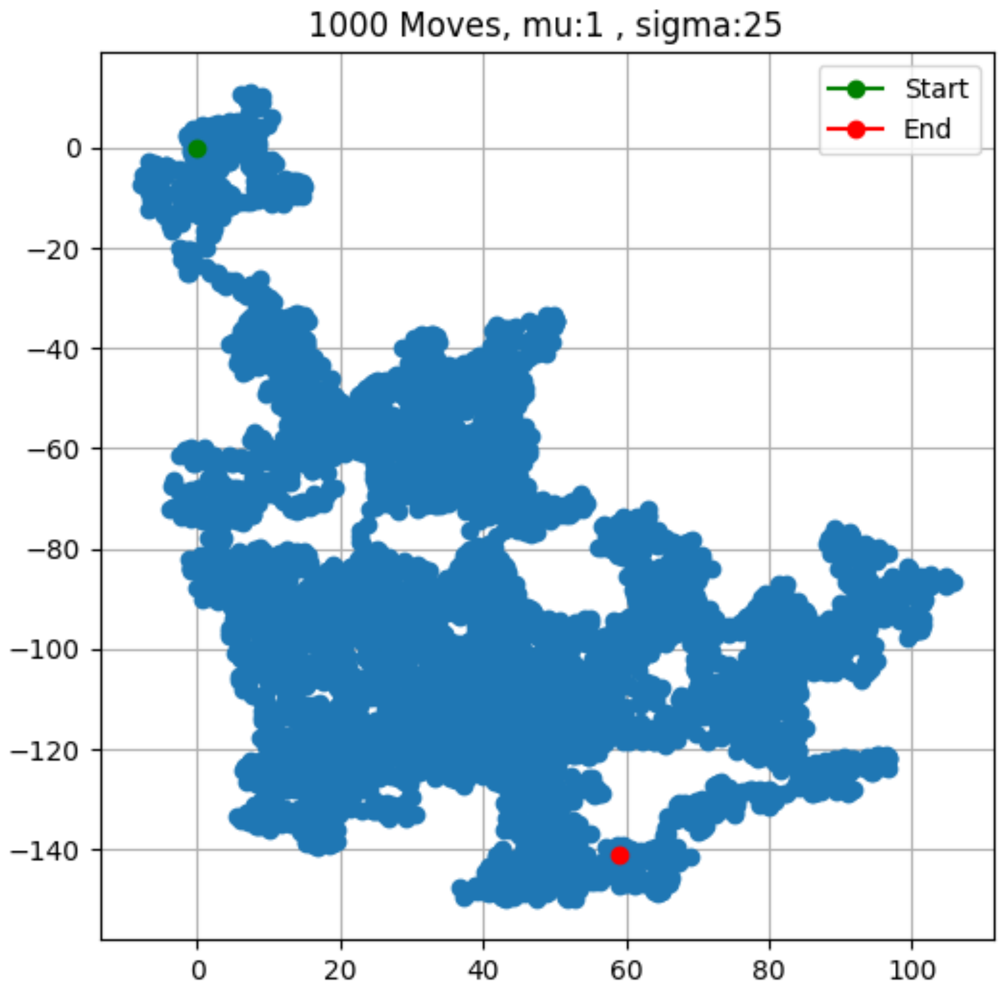


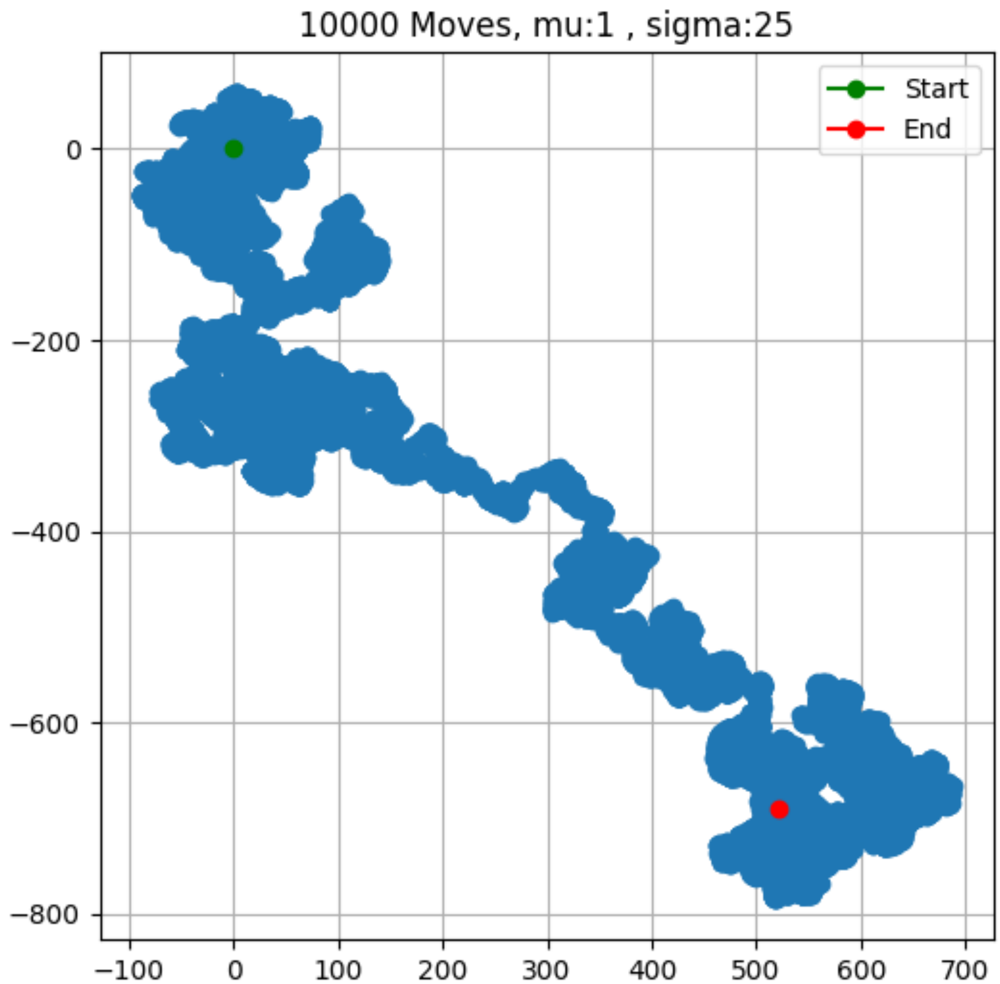
100000 Moves, mu:1 , sigma:20
Start
End
800
600
400
200
0
−200
−400
−600
0
200
400
600
800
1000
1200
1400

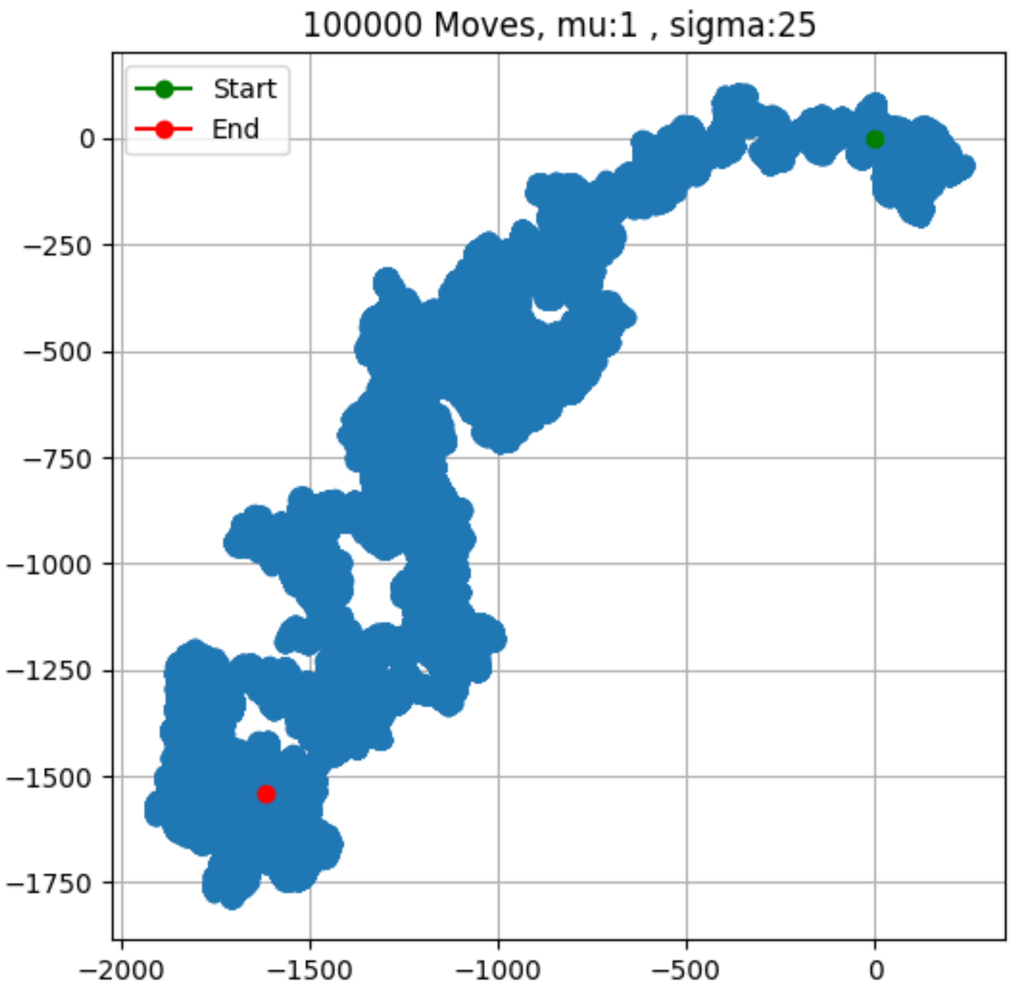


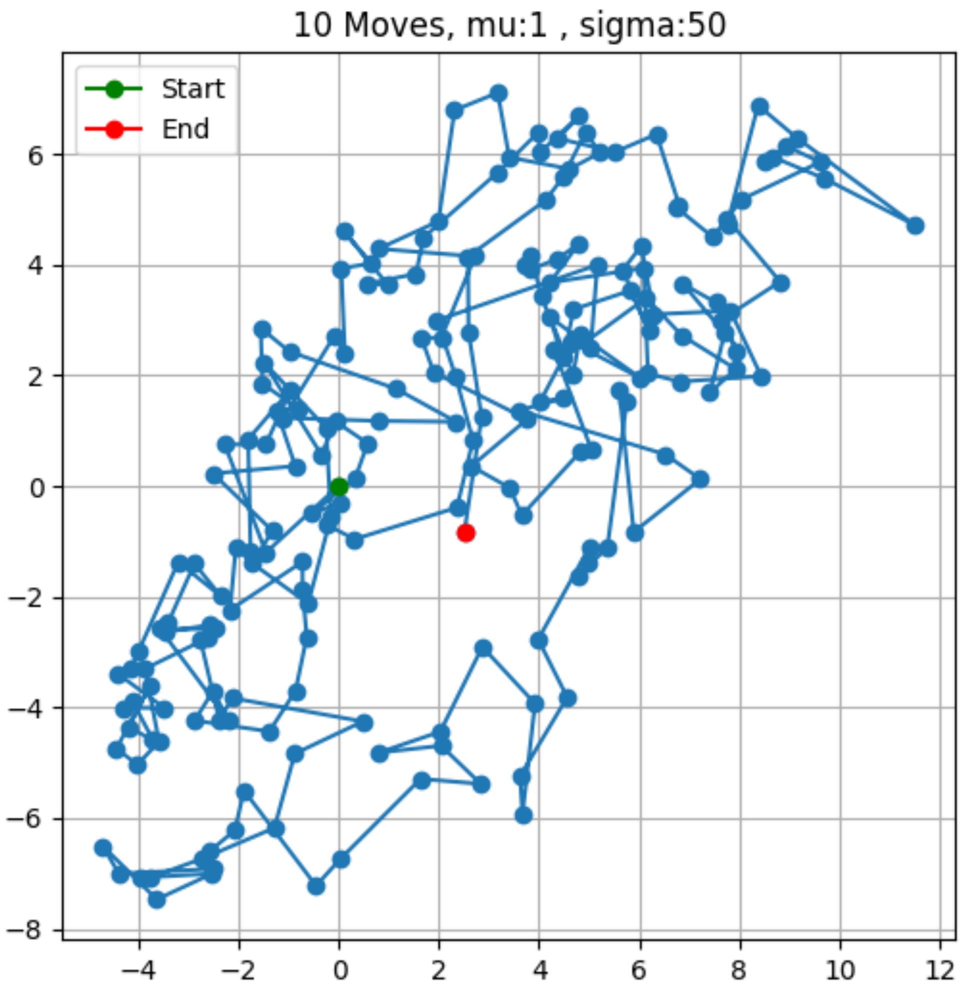
10 Moves, mu:1 , sigma:25
Start
End
0
−5
−10
−15
−20
−15.0
−12.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5

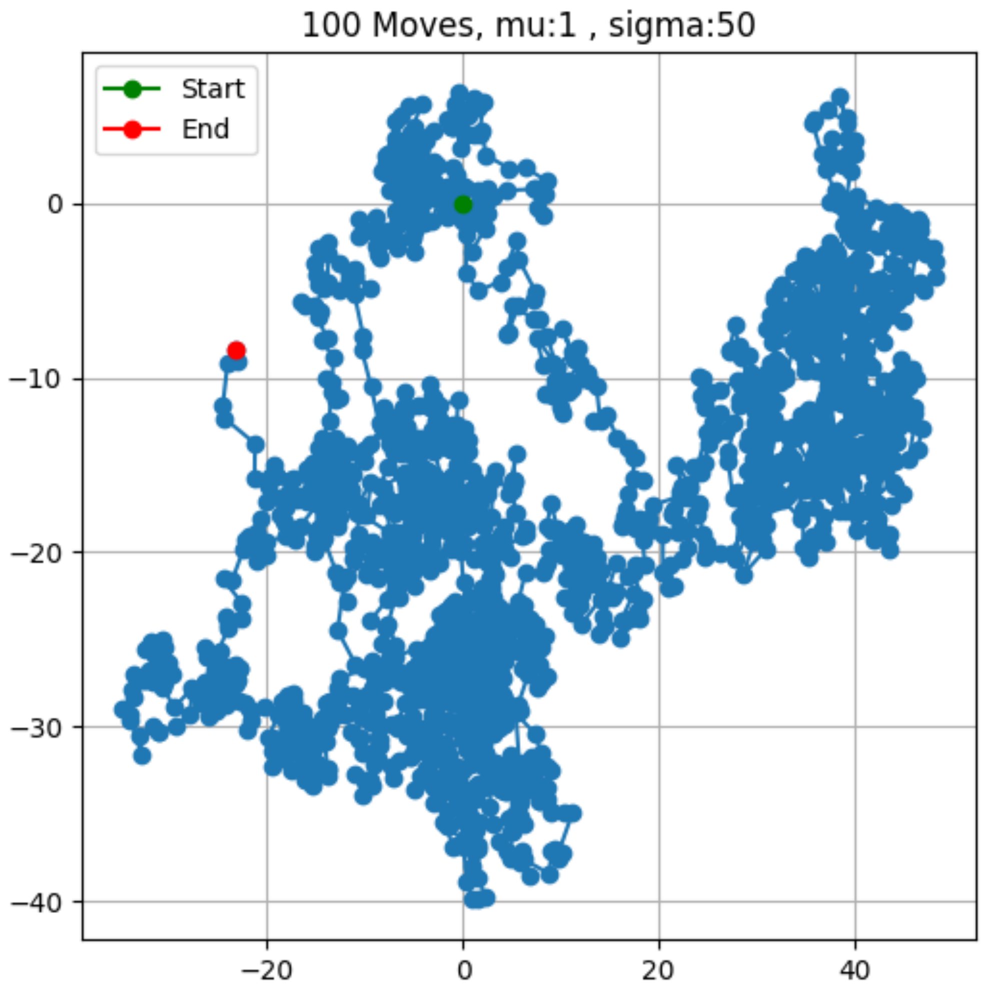

100 Moves, mu:1 , sigma:25
Start
End
0
−5
−10
−15
−20
−25
−30
−5
0
5
10
15
20

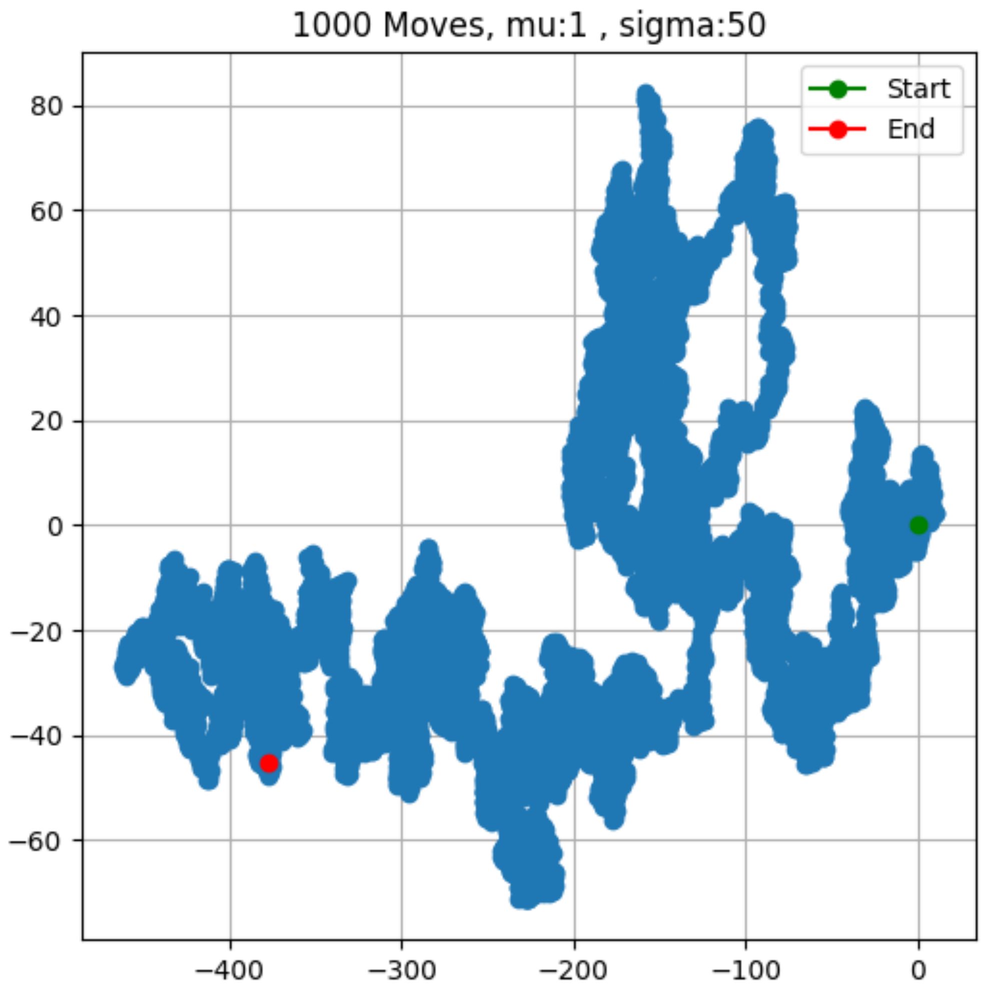

1000 Moves, mu:1 , sigma:25
Start
End
0
−20
−40
−60
−80
−100
−120
−140
0
20
40
60
80
100

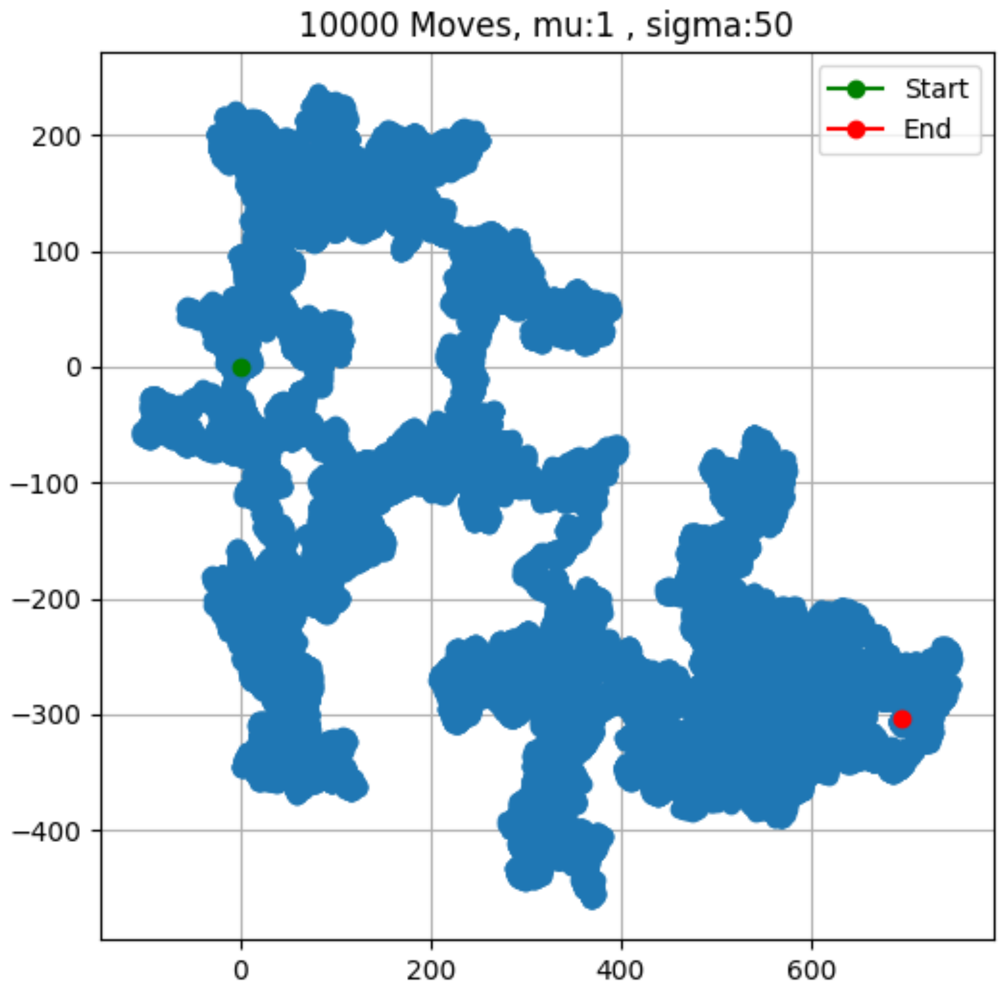

10000 Moves, mu:1 , sigma:25
Start
End
0
−200
−400
−600
−800
−100
0
100
200
300
400
500
600
700

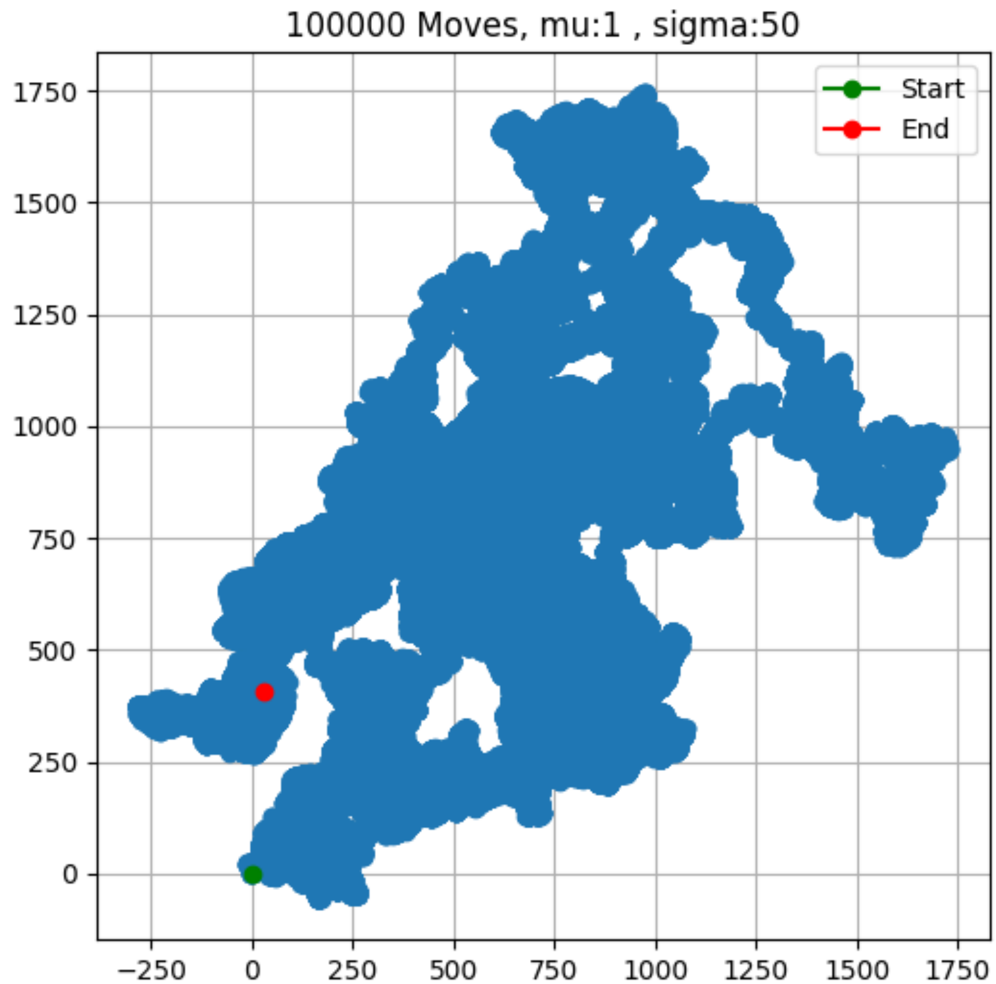

100000 Moves, mu:1 , sigma:25
Start
End
0
−250
−500
−750
−1000
−1250
−1500
−1750
−2000
−1500
−1000
−500
0

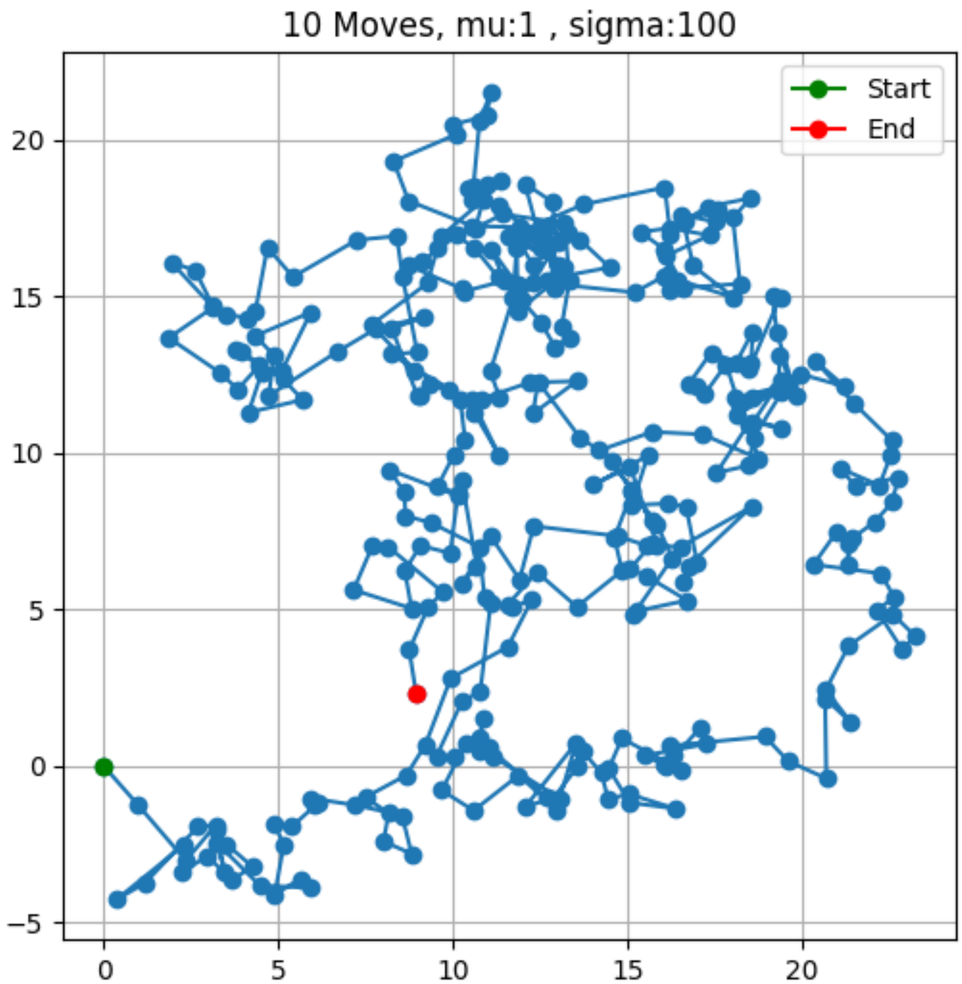

10 Moves, mu:1 , sigma:50
Start
End
6
4
2
0
−2
−4
−6
−8
−4
−2
0
2
4
6
8
10
12

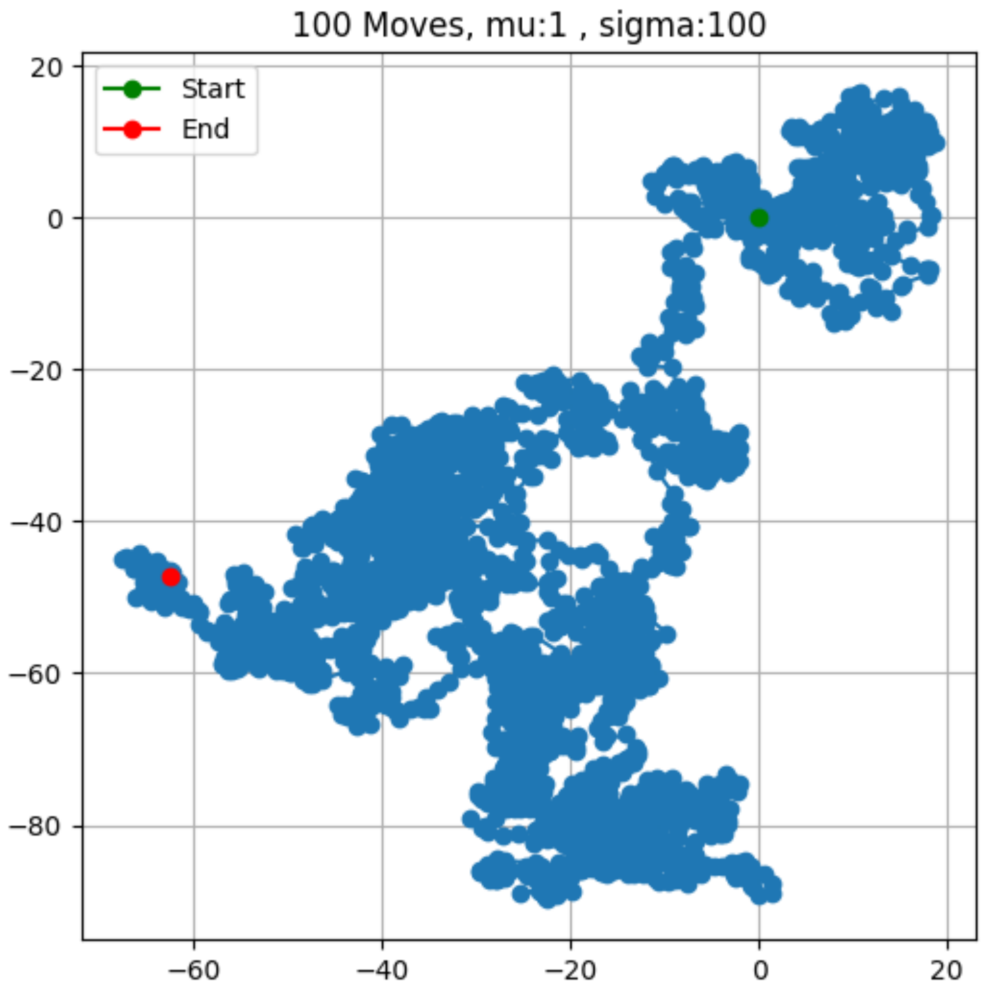

100 Moves, mu:1 , sigma:50
Start
End
0
−10
−20
−30
−40
−20
0
20
40

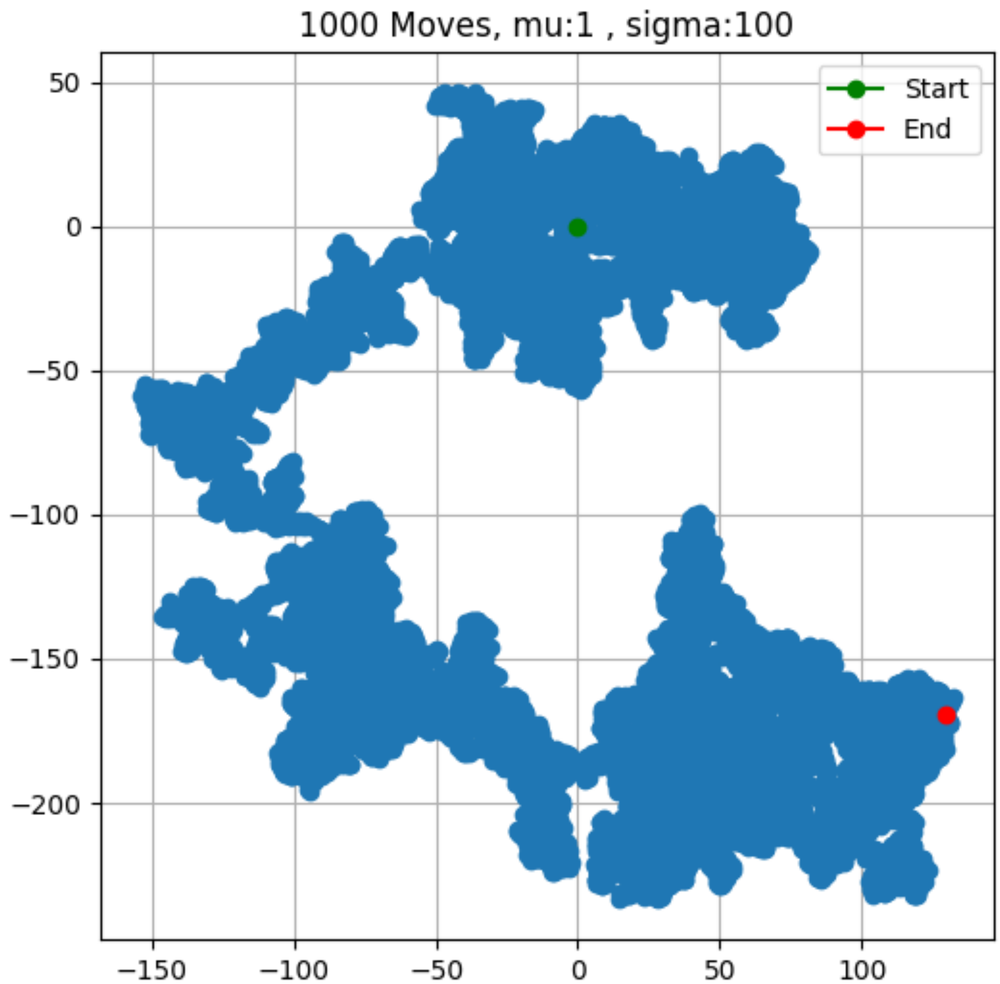

1000 Moves, mu:1 , sigma:50
Start
End
80
60
40
20
0
−20
−40
−60
−400
−300
−200
−100
0

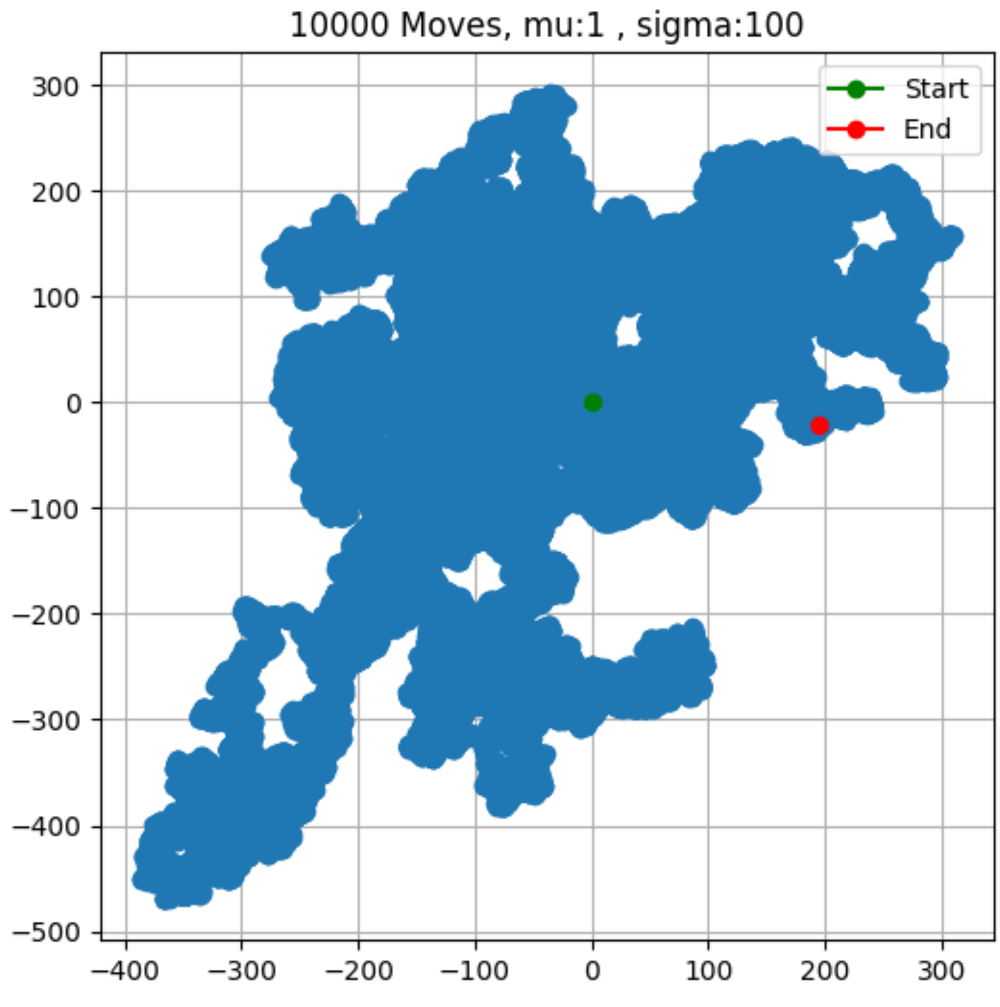

10000 Moves, mu:1 , sigma:50
Start
End
200
100
0
−100
−200
−300
−400
0
200
400
600

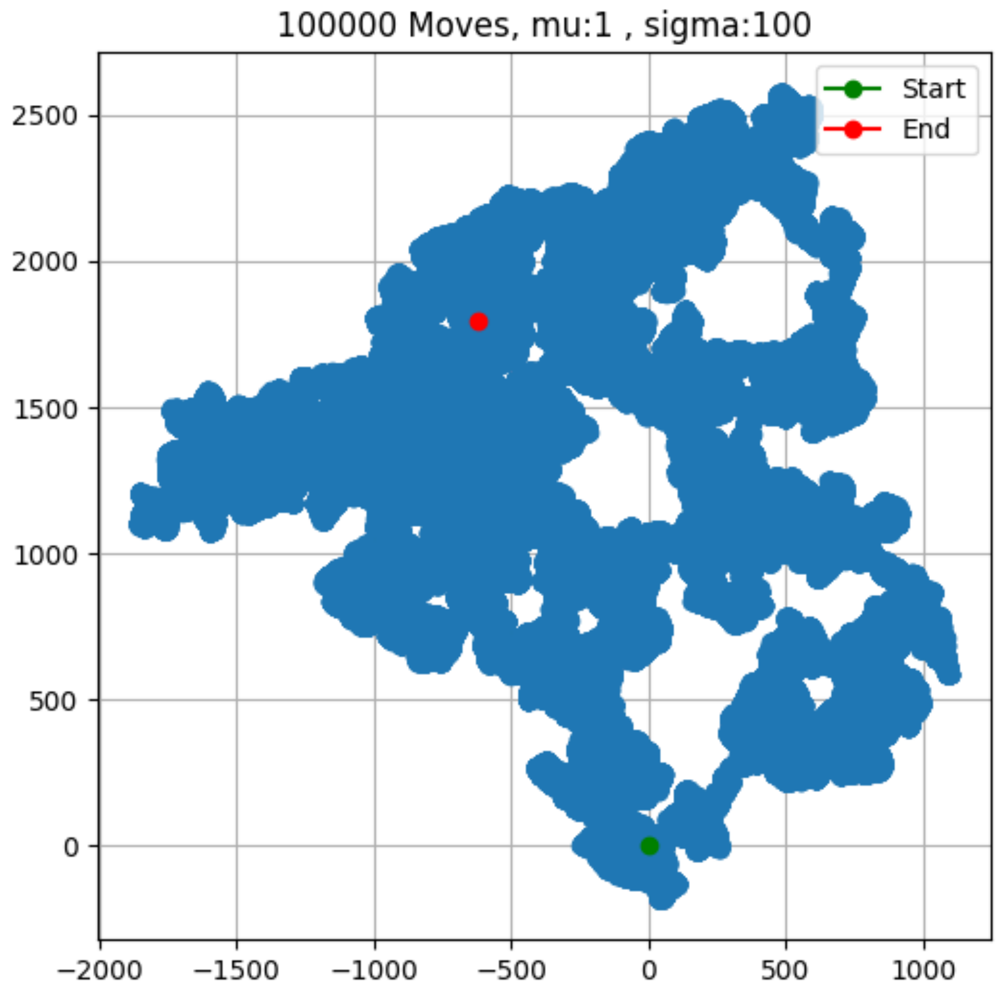

100000 Moves, mu:1 , sigma:50
Start
End
1750
1500
1250
1000
750
500
250
0
−250
0
250
500
750
1000
1250
1500
1750

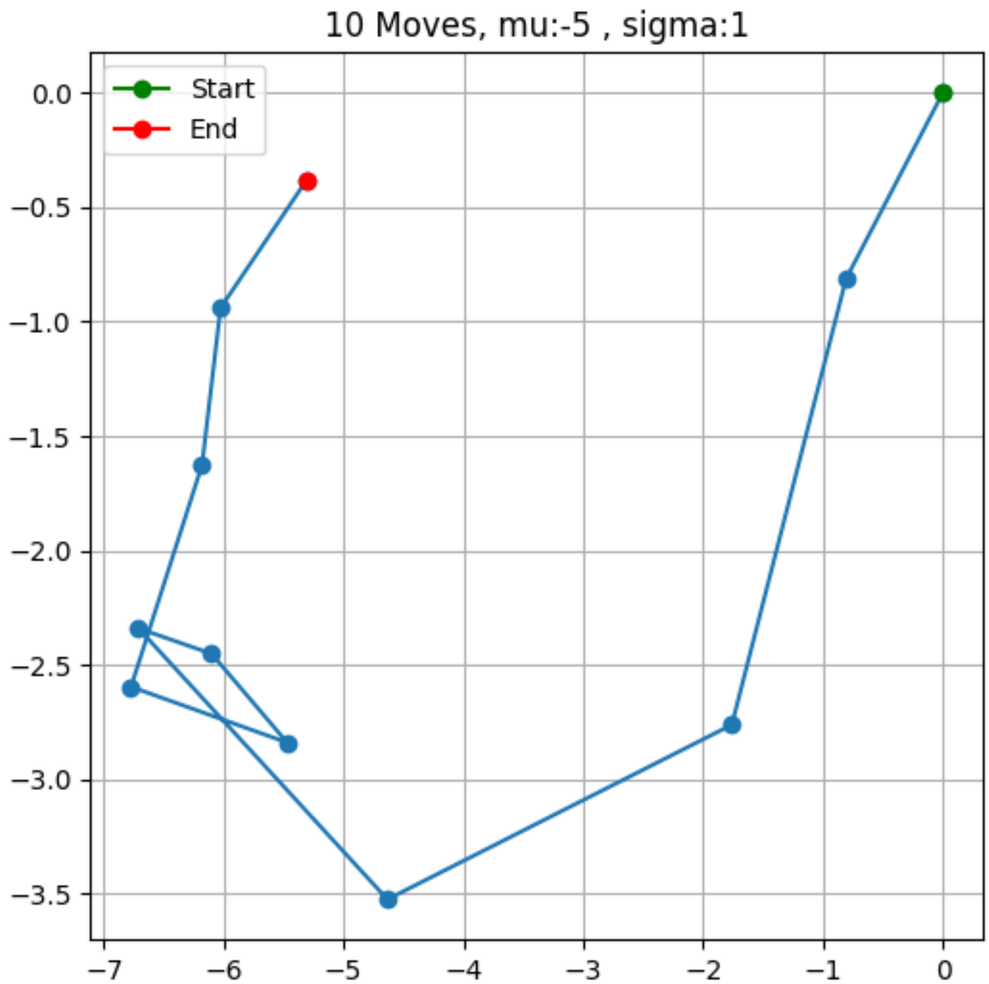

10 Moves, mu:1 , sigma:100
Start
End
20
15
10
5
0
−5
0
5
10
15
20

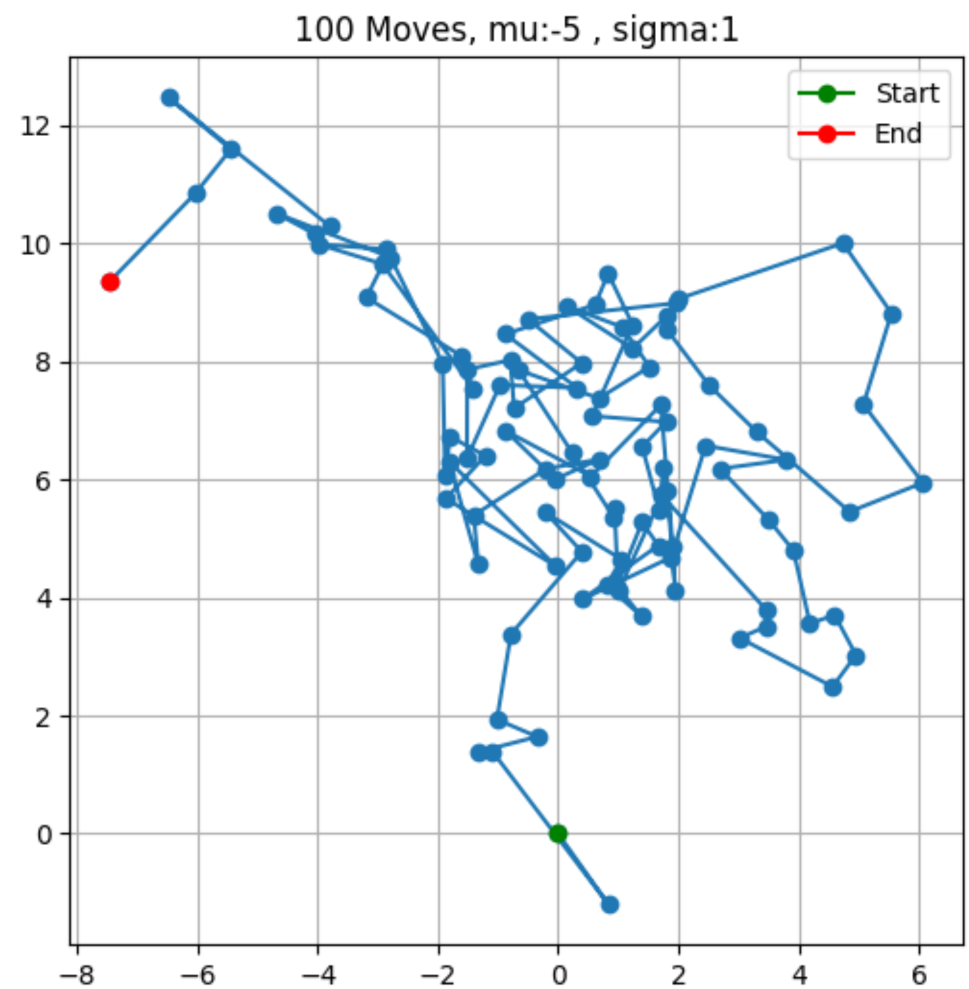

100 Moves, mu:1 , sigma:100
Start
End
20
0
−20
−40
−60
−80
−60
−40
−20
0
20

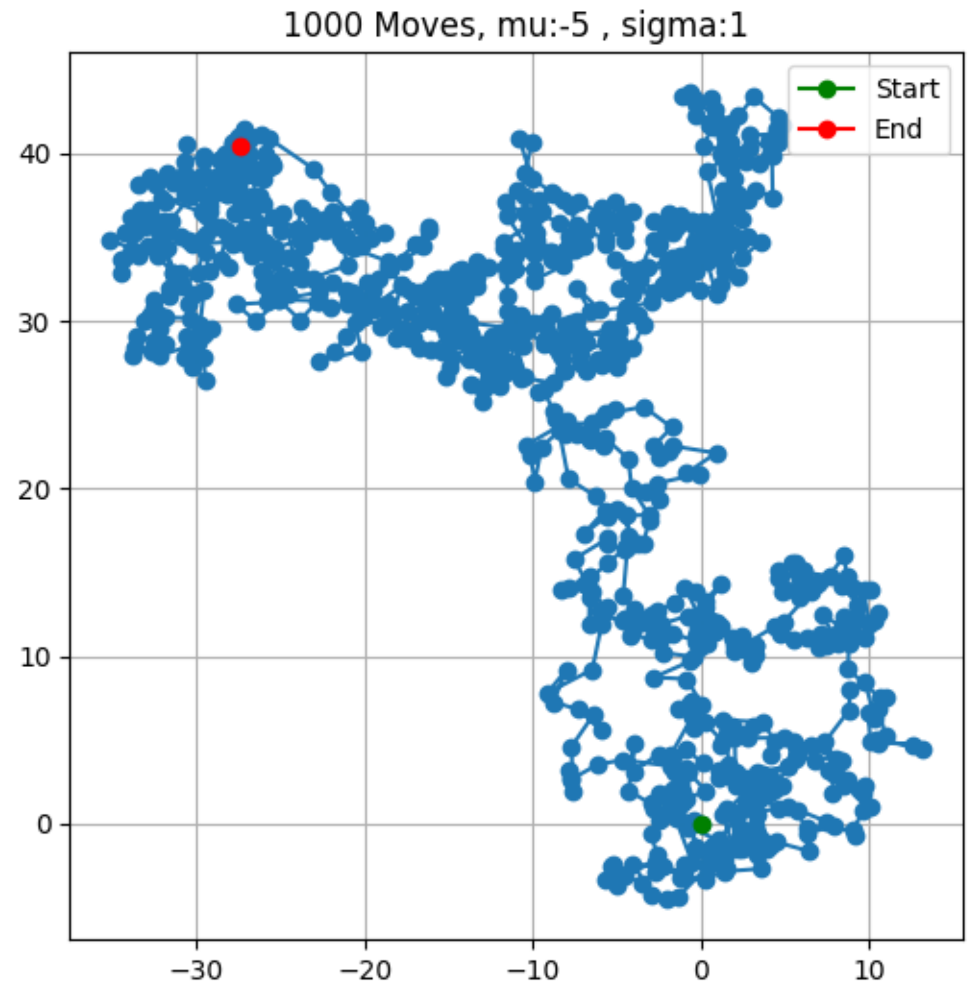

1000 Moves, mu:1 , sigma:100
Start
End
50
0
−50
−100
−150
−200
−150
−100
−50
0
50
100

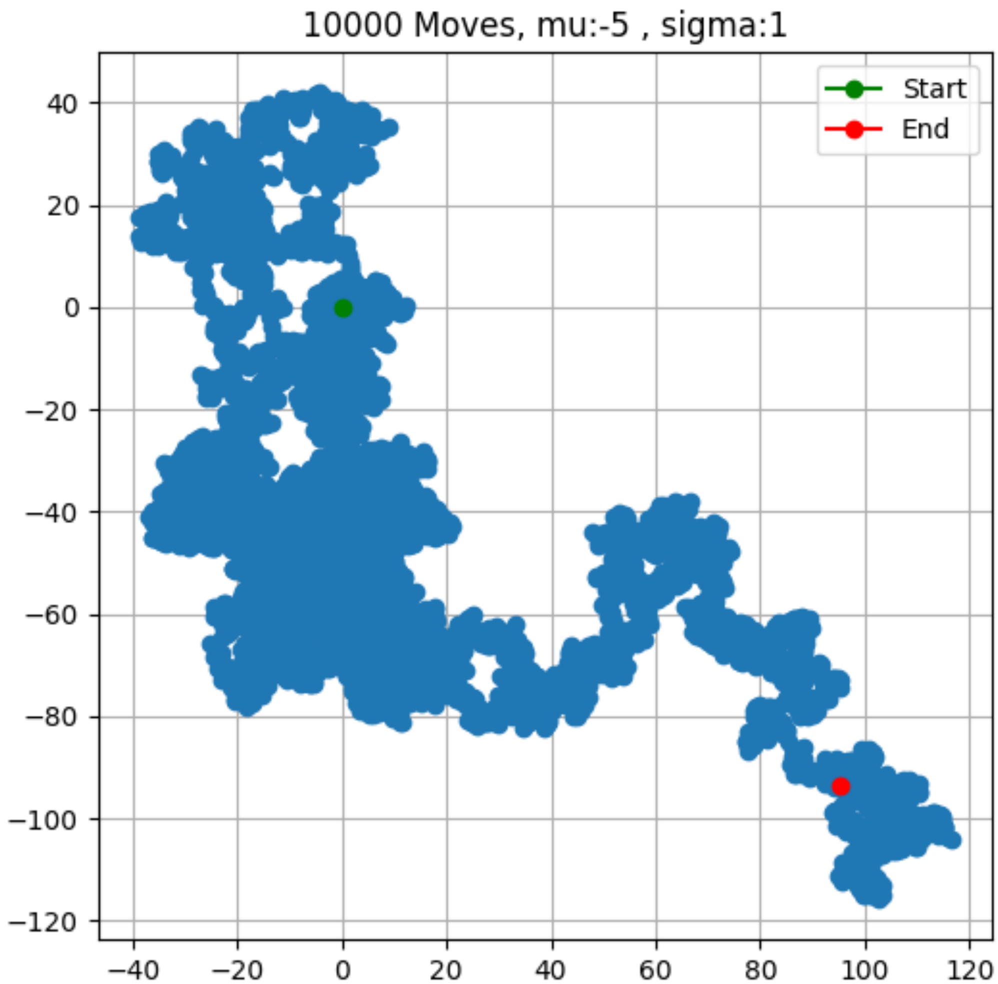

10000 Moves, mu:1 , sigma:100
Start
End
300
200
100
0
−100
−200
−300
−400
−500
−400
−300
−200
−100
0
100
200
300

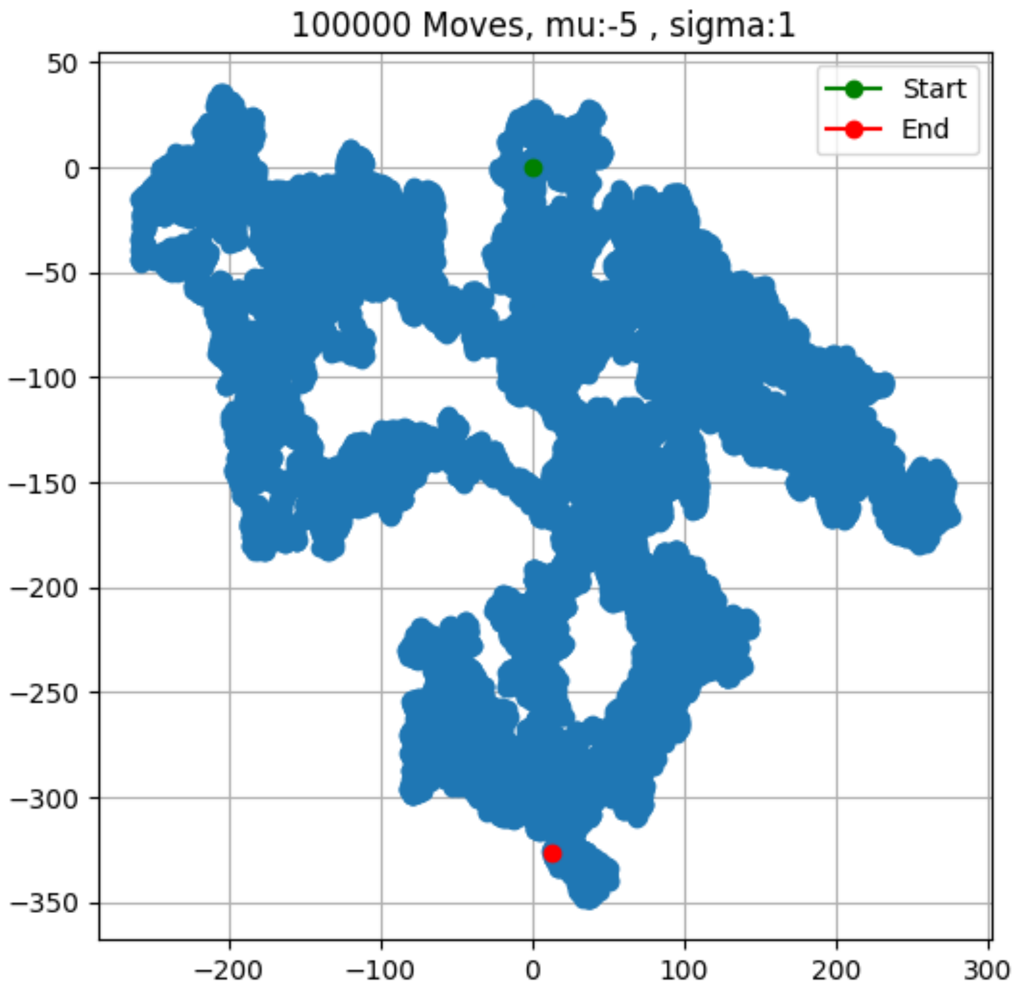

100000 Moves, mu:1 , sigma:100
Start
End
2500
2000
1500
1000
500
0
−2000
−1500
−1000
−500
0
500
1000

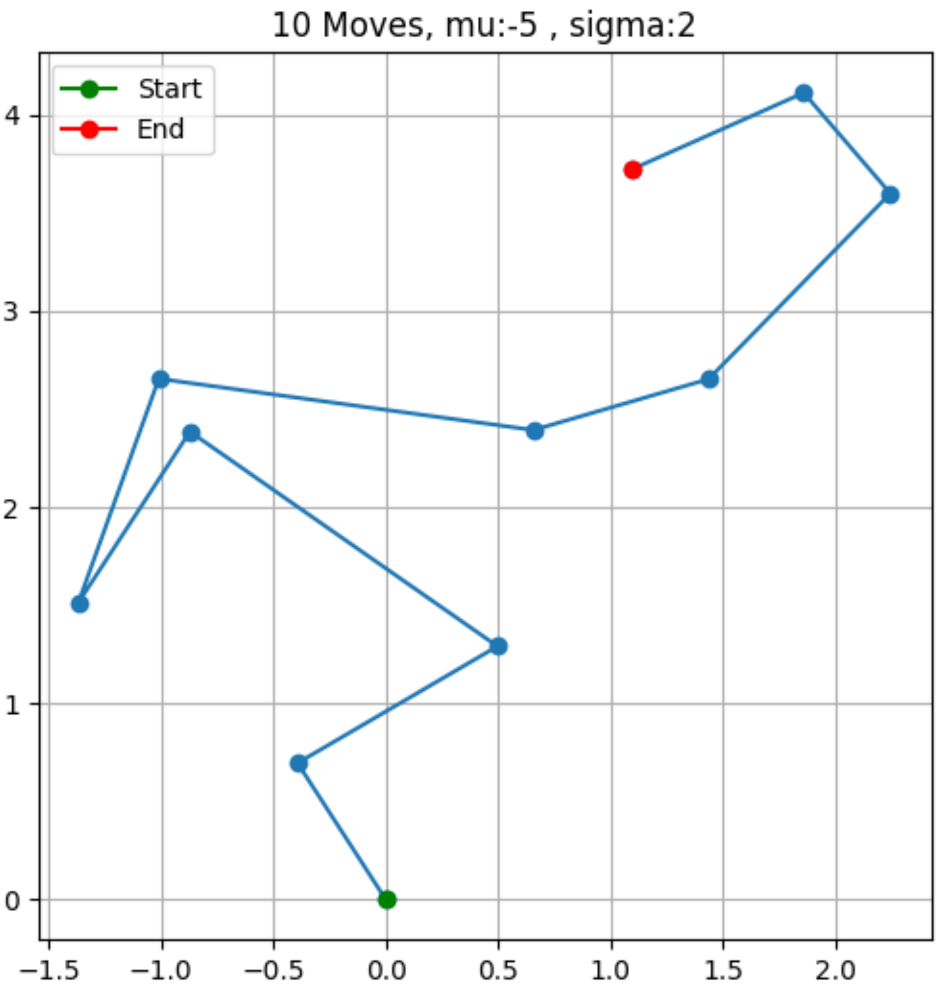

10 Moves, mu:-5 , sigma:1
Start
End
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
−3.5
−7
−6
−5
−4
−3
−2
−1
0

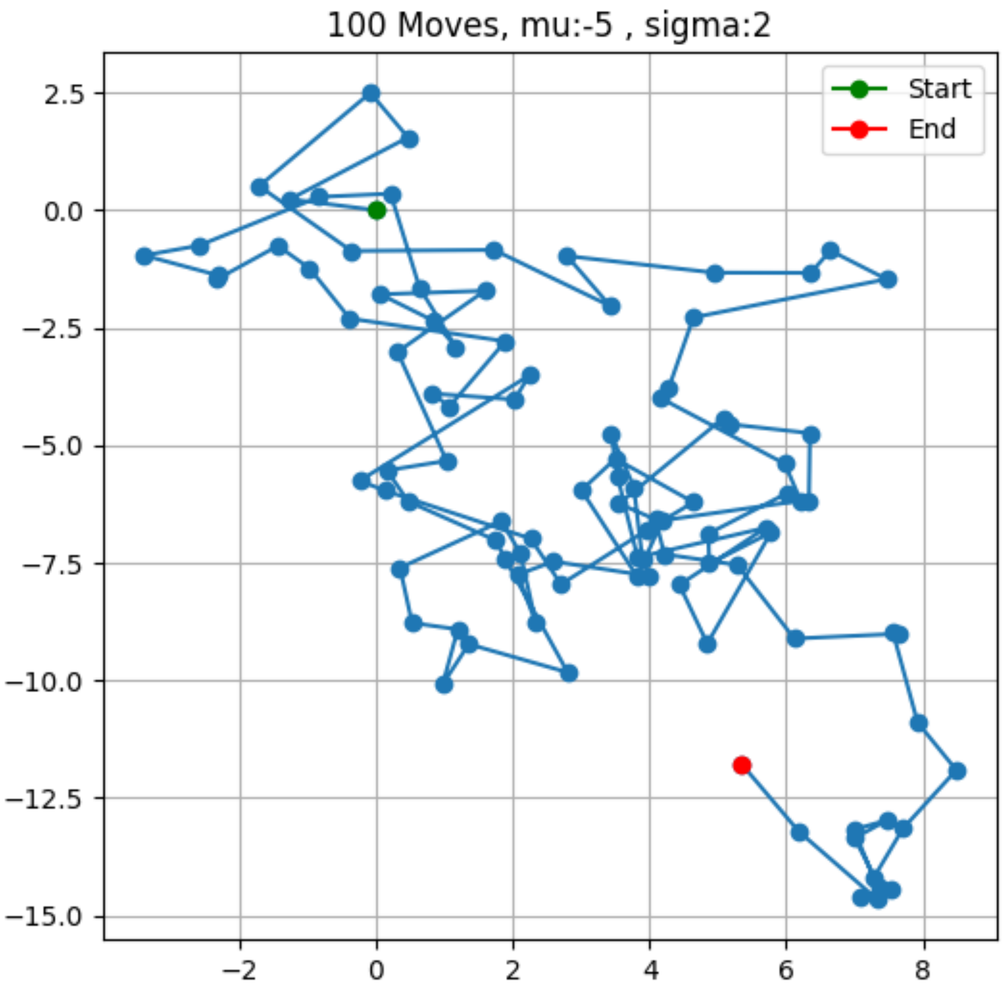

100 Moves, mu:-5 , sigma:1
Start
End
12
10
8
6
4
2
0
−8
−6
−4
−2
0
2
4
6

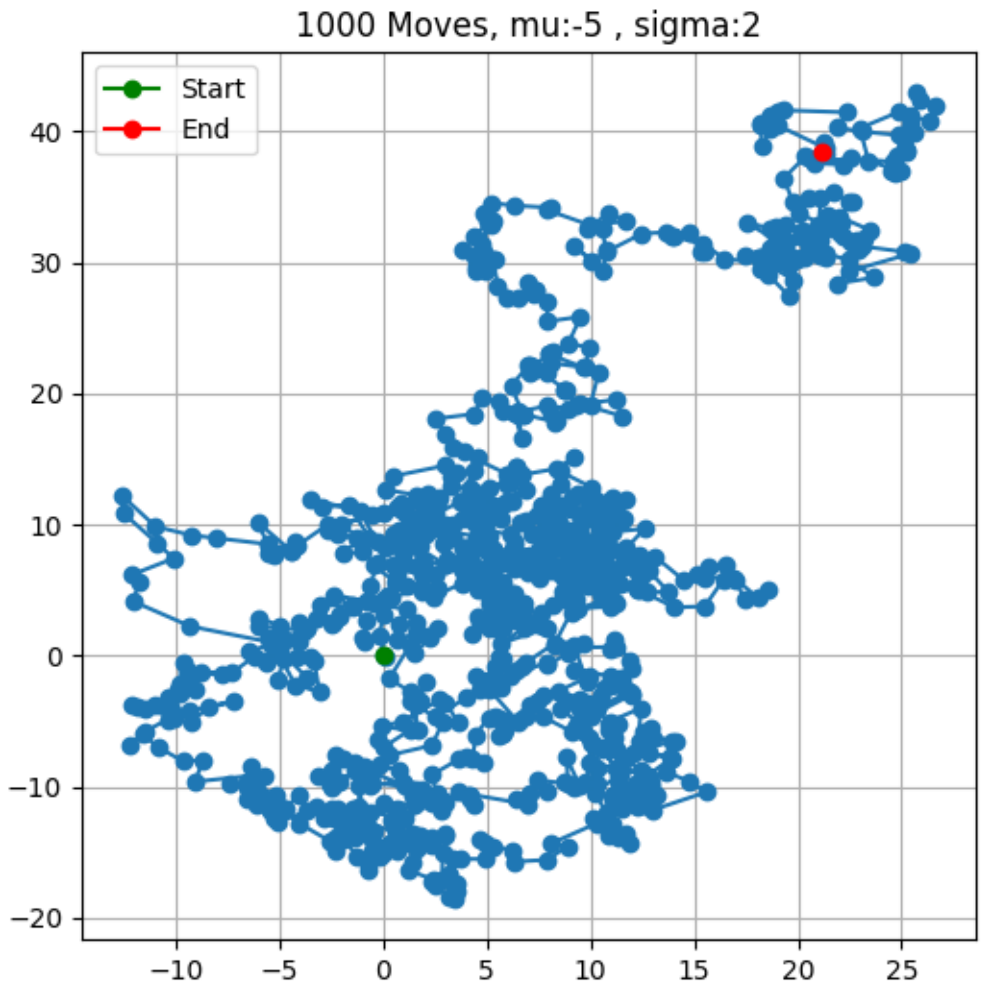

1000 Moves, mu:-5 , sigma:1
Start
End
40
30
20
10
0
−30
−20
−10
0
10

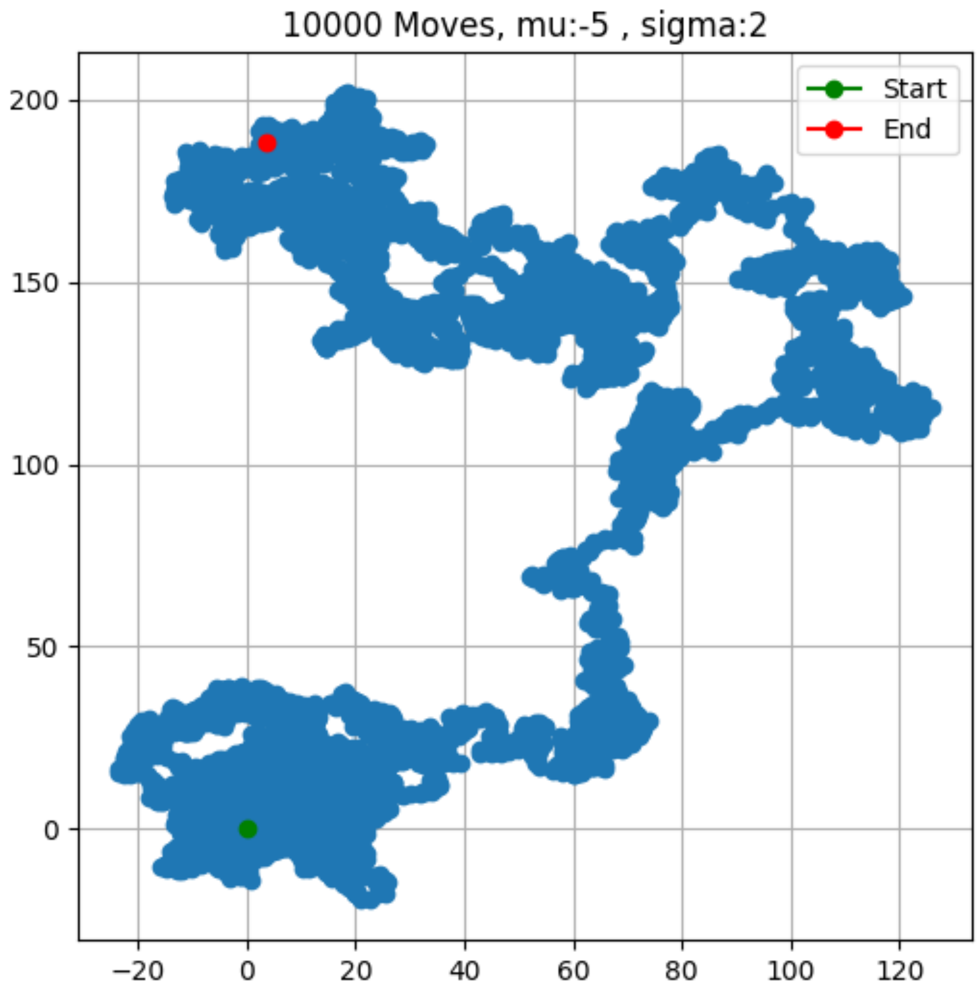

10000 Moves, mu:-5 , sigma:1
Start
End
40
20
0
−20
−40
−60
−80
−100
−120
−40
−20
0
20
40
60
80
100
120

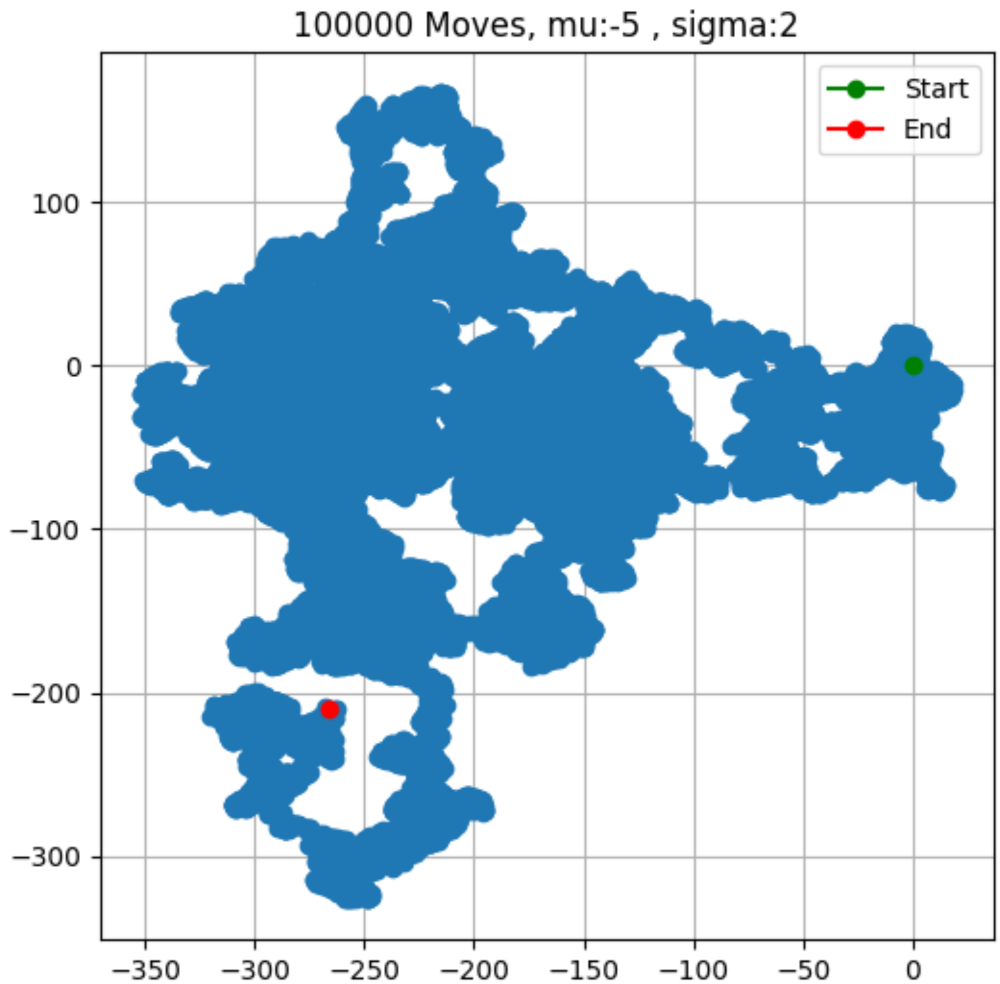

100000 Moves, mu:-5 , sigma:1
Start
End
50
0
−50
−100
−150
−200
−250
−300
−350
−200
−100
0
100
200
300

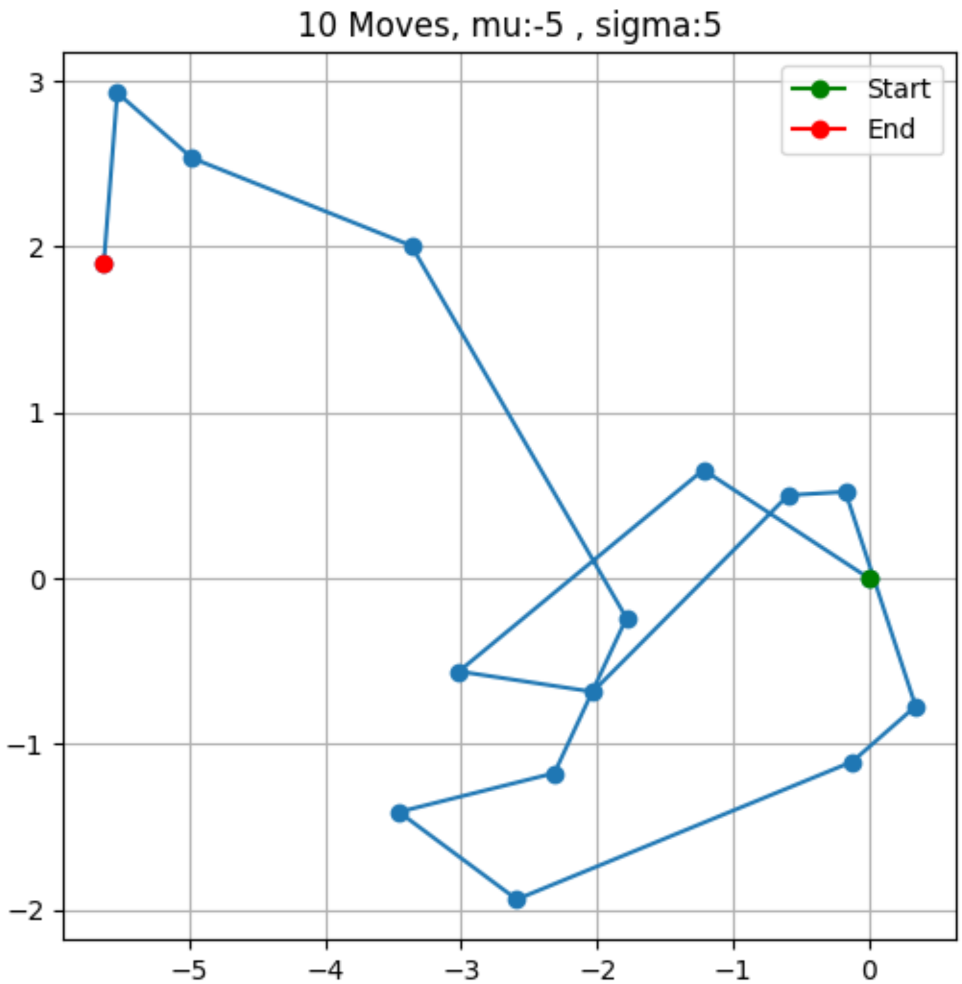

10 Moves, mu:-5 , sigma:2
Start
End
4
3
2
1
0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


100 Moves, mu:-5 , sigma:2
Start
End
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−12.5
−15.0
−2
0
2
4
6
8

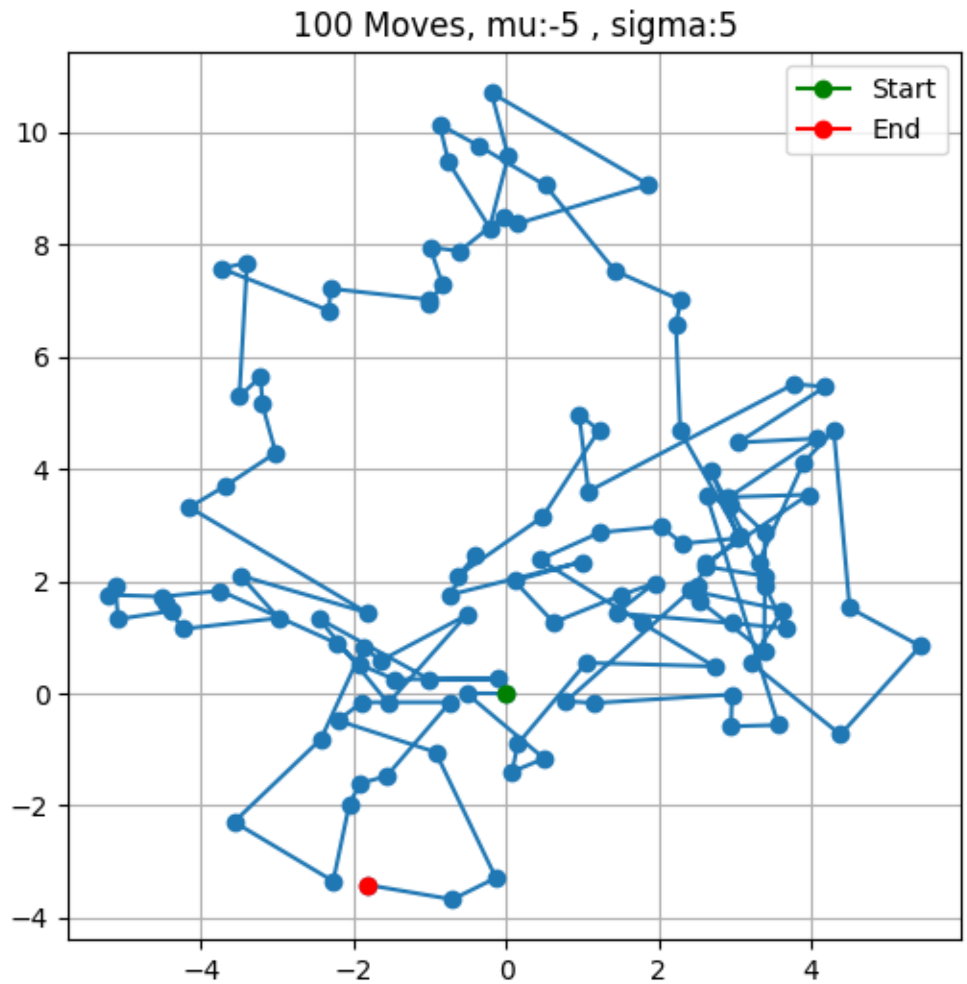

1000 Moves, mu:-5 , sigma:2
Start
End
40
30
20
10
0
−10
−20
−10
−5
0
5
10
15
20
25

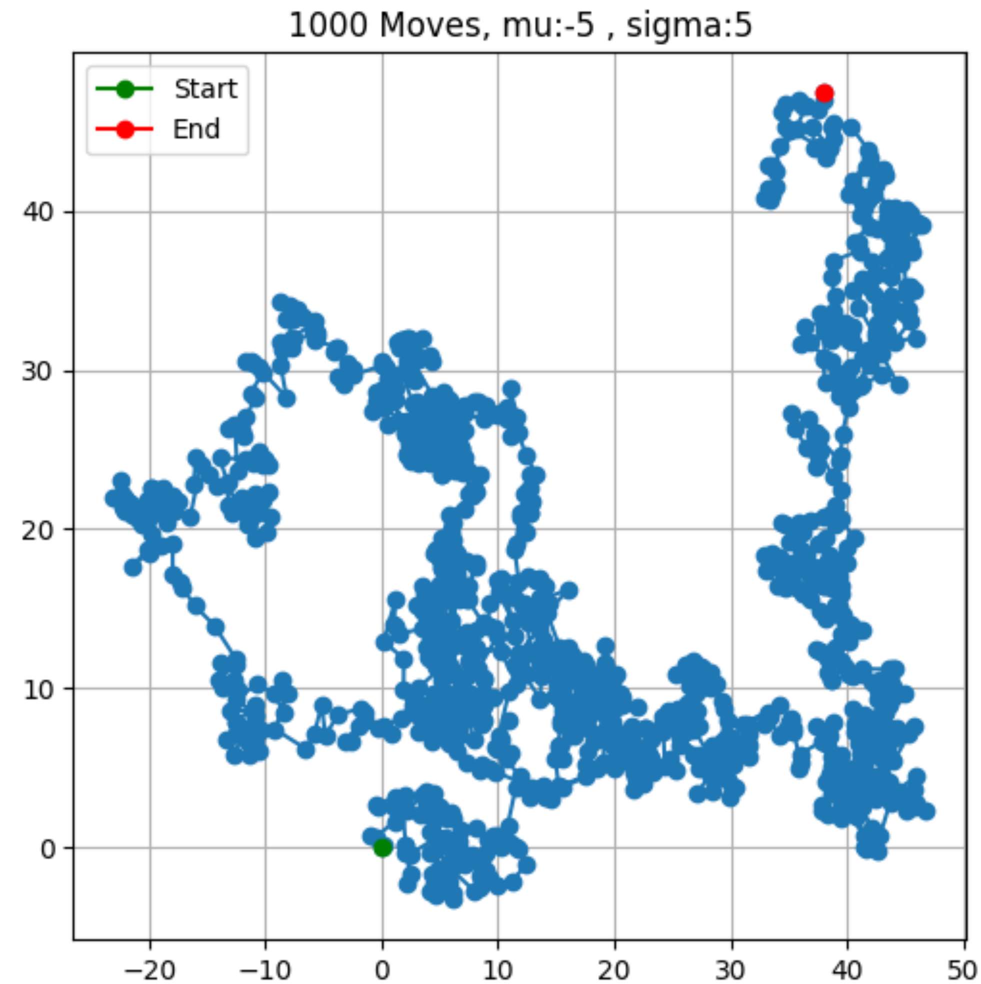

10000 Moves, mu:-5 , sigma:2
Start
End
200
150
100
50
0
−20
0
20
40
60
80
100
120

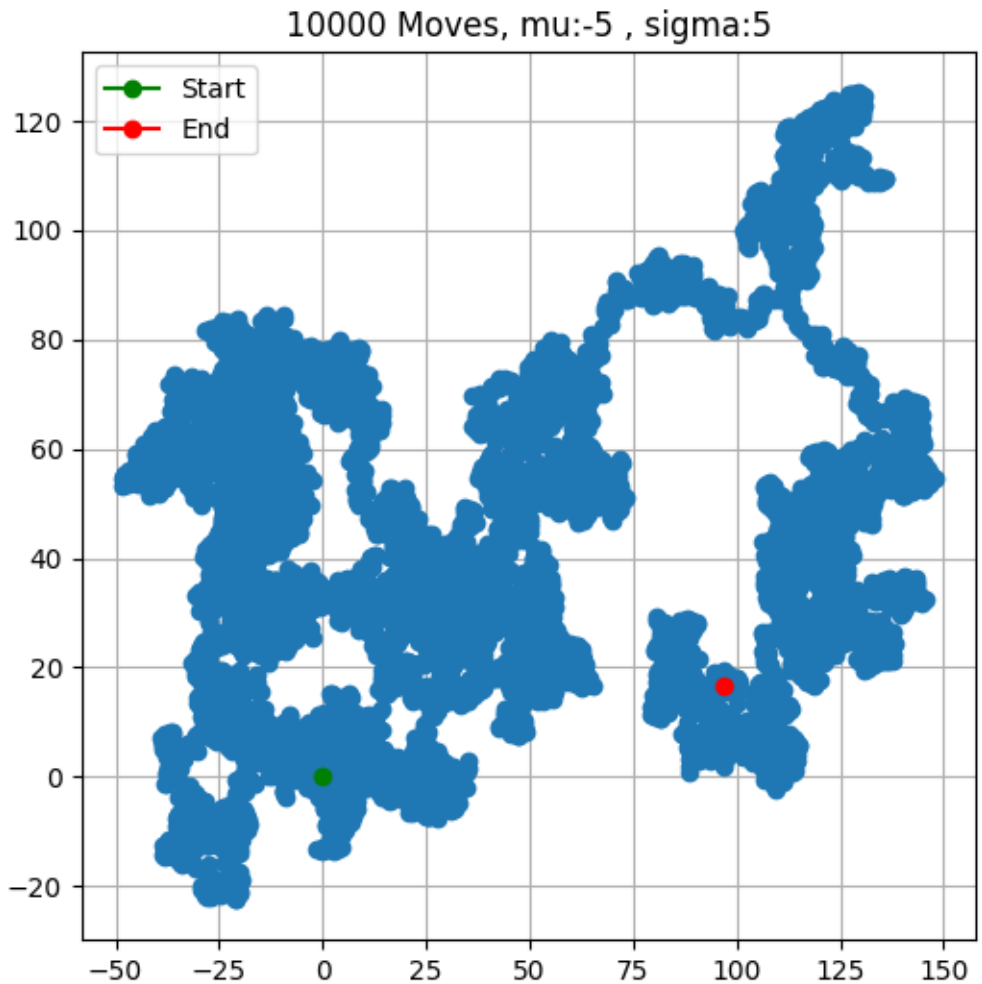

100000 Moves, mu:-5 , sigma:2
Start
End
100
0
−100
−200
−300
−350
−300
−250
−200
−150
−100
−50
0

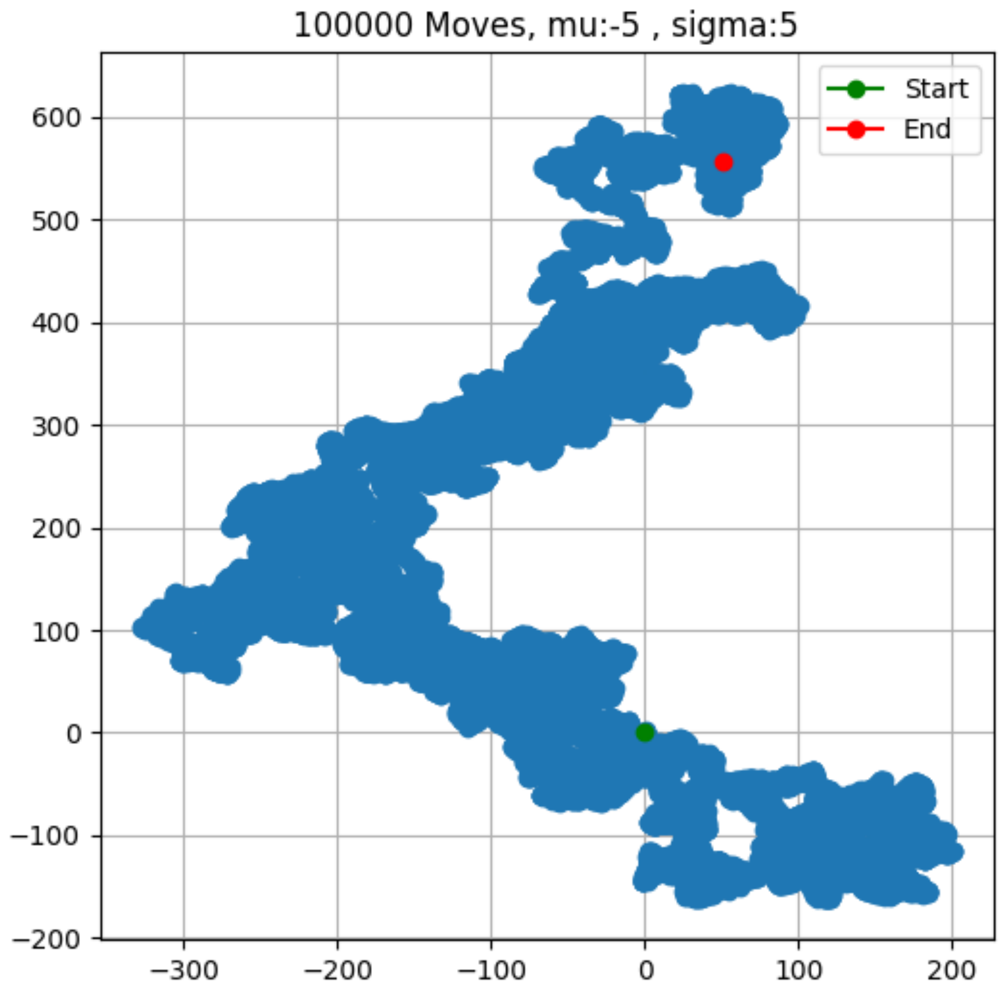


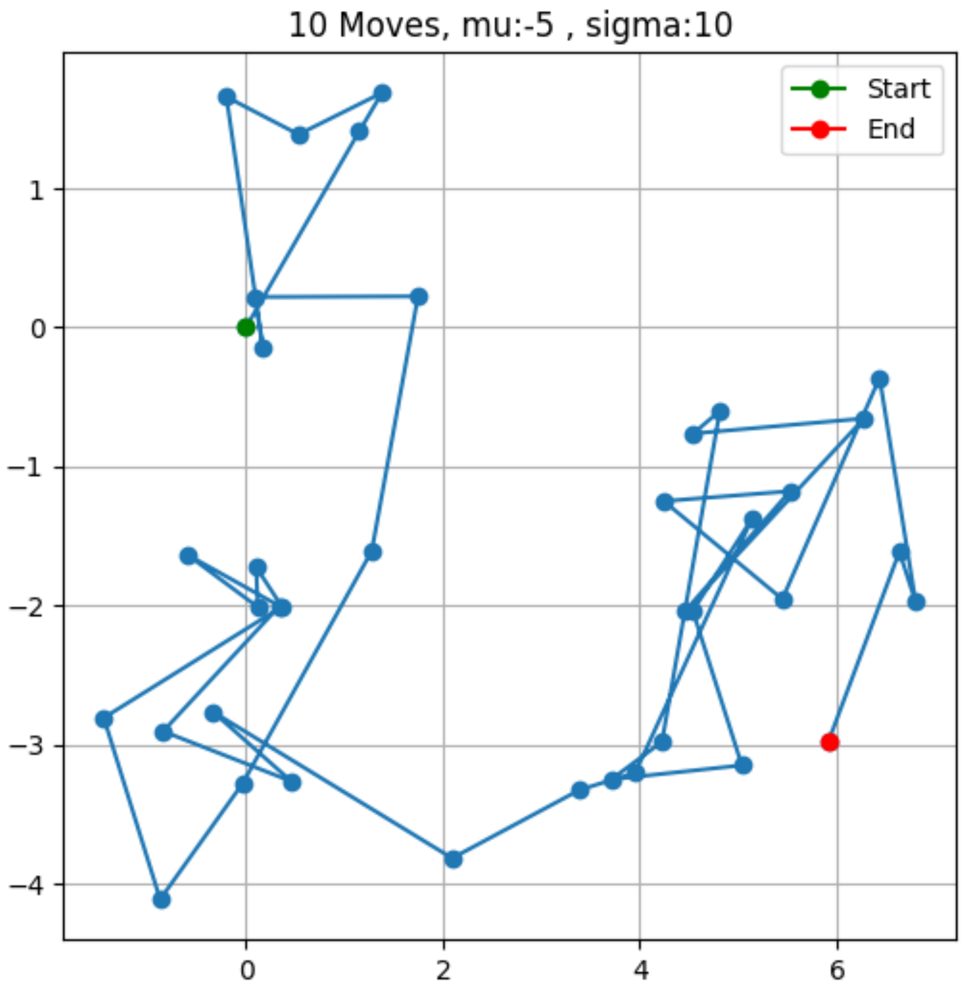
10 Moves, mu:-5 , sigma:5
Start
End
3
2
1
0
−1
−2
−5
−4
−3
−2
−1
0

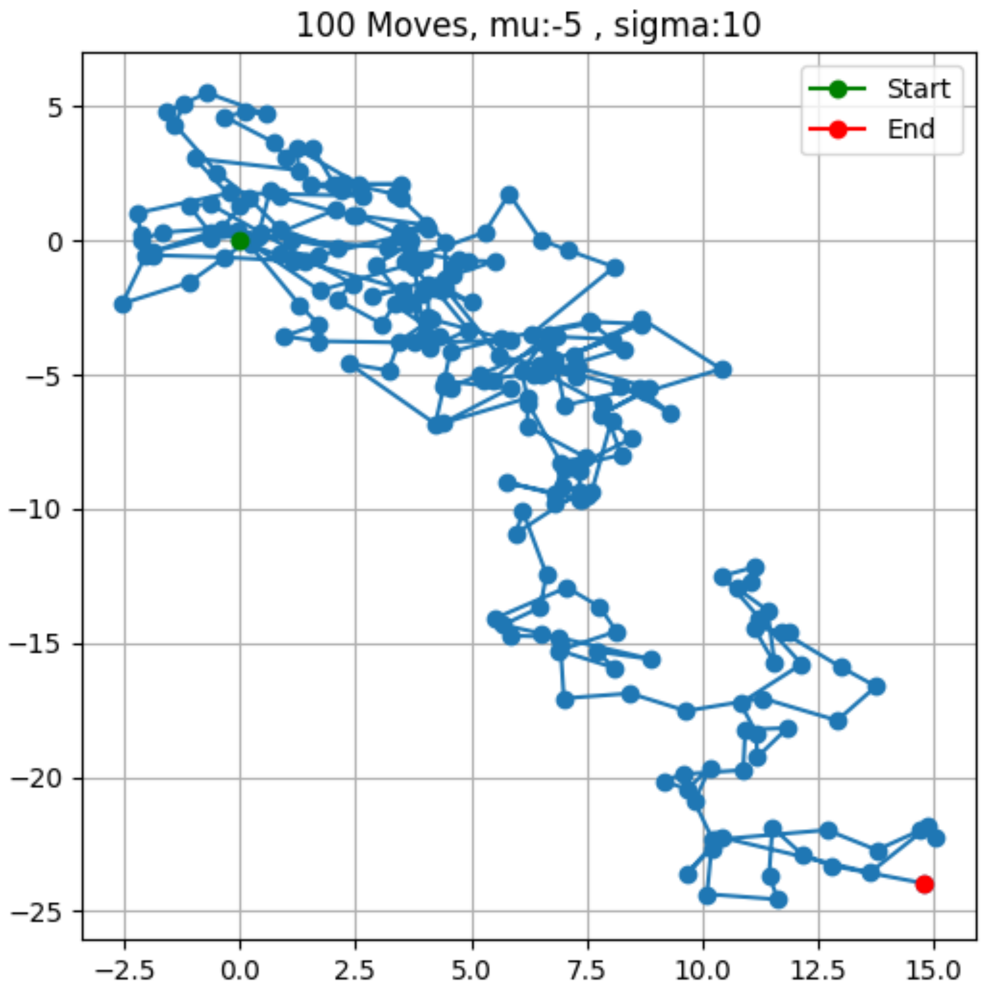

100 Moves, mu:-5 , sigma:5
Start
End
10
8
6
4
2
0
−2
−4
−4
−2
0
2
4

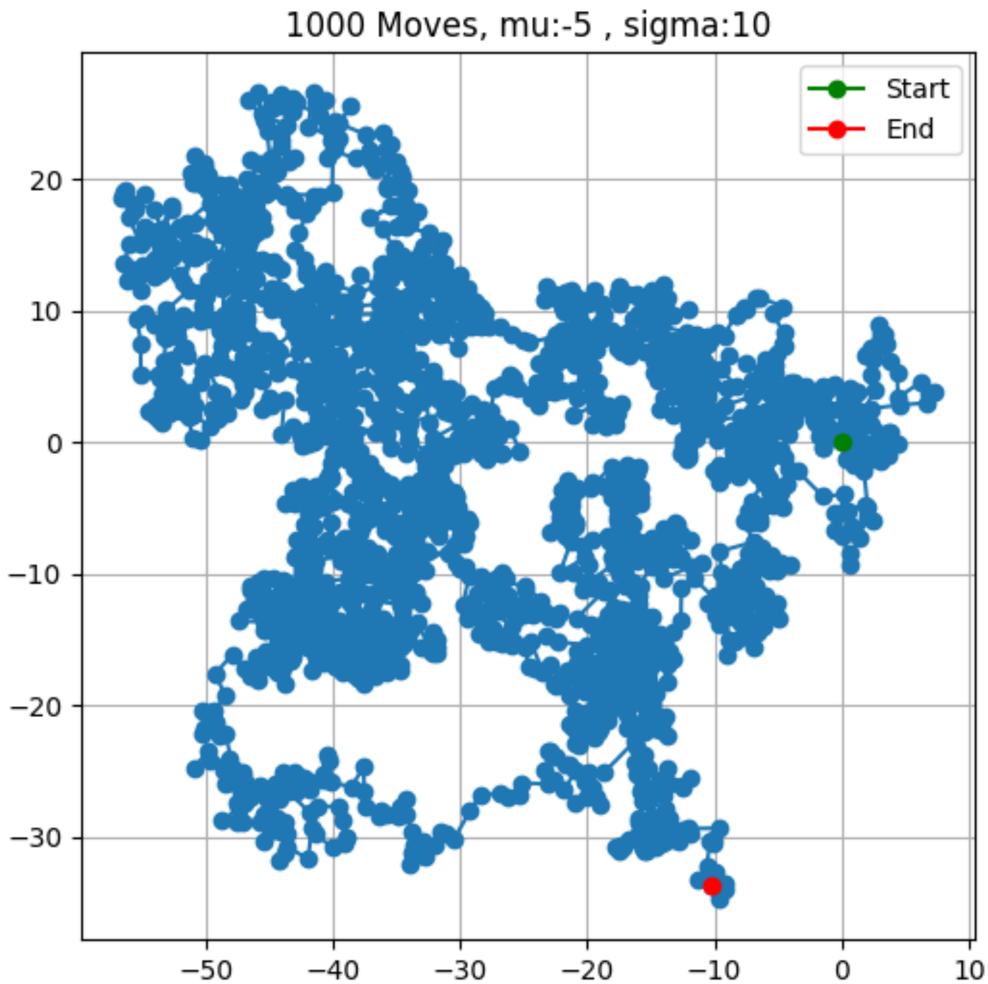

1000 Moves, mu:-5 , sigma:5
Start
End
40
30
20
10
0
−20
−10
0
10
20
30
40
50

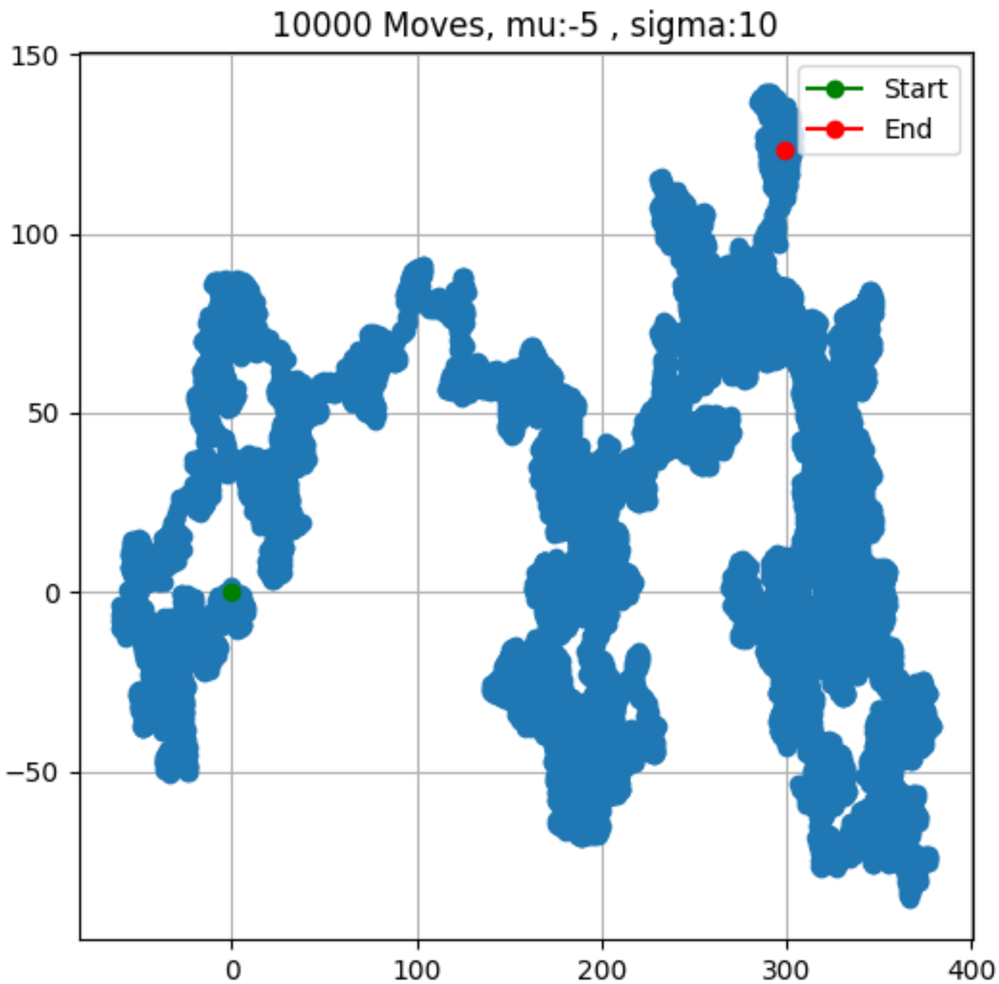

10000 Moves, mu:-5 , sigma:5
Start
End
120
100
80
60
40
20
0
−20
−50
−25
0
25
50
75
100
125
150

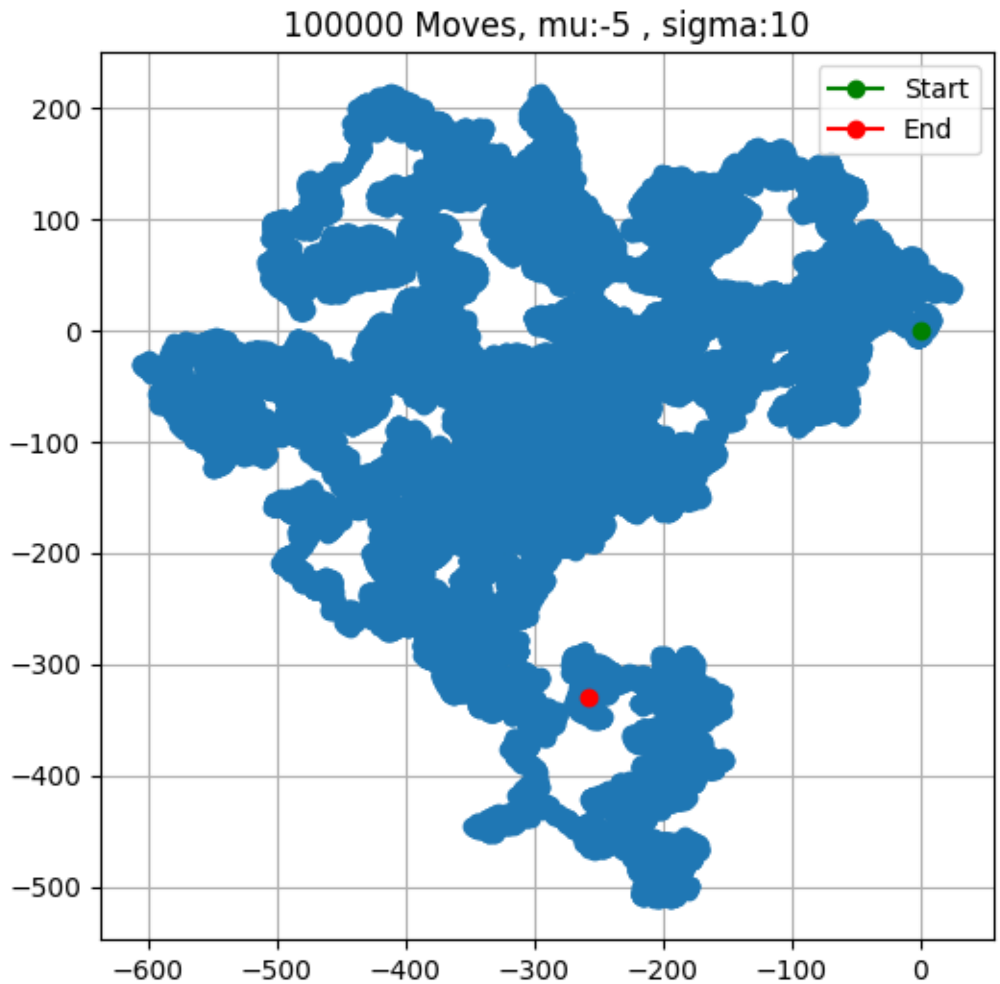

100000 Moves, mu:-5 , sigma:5
Start
End
600
500
400
300
200
100
0
−100
−200
−300
−200
−100
0
100
200

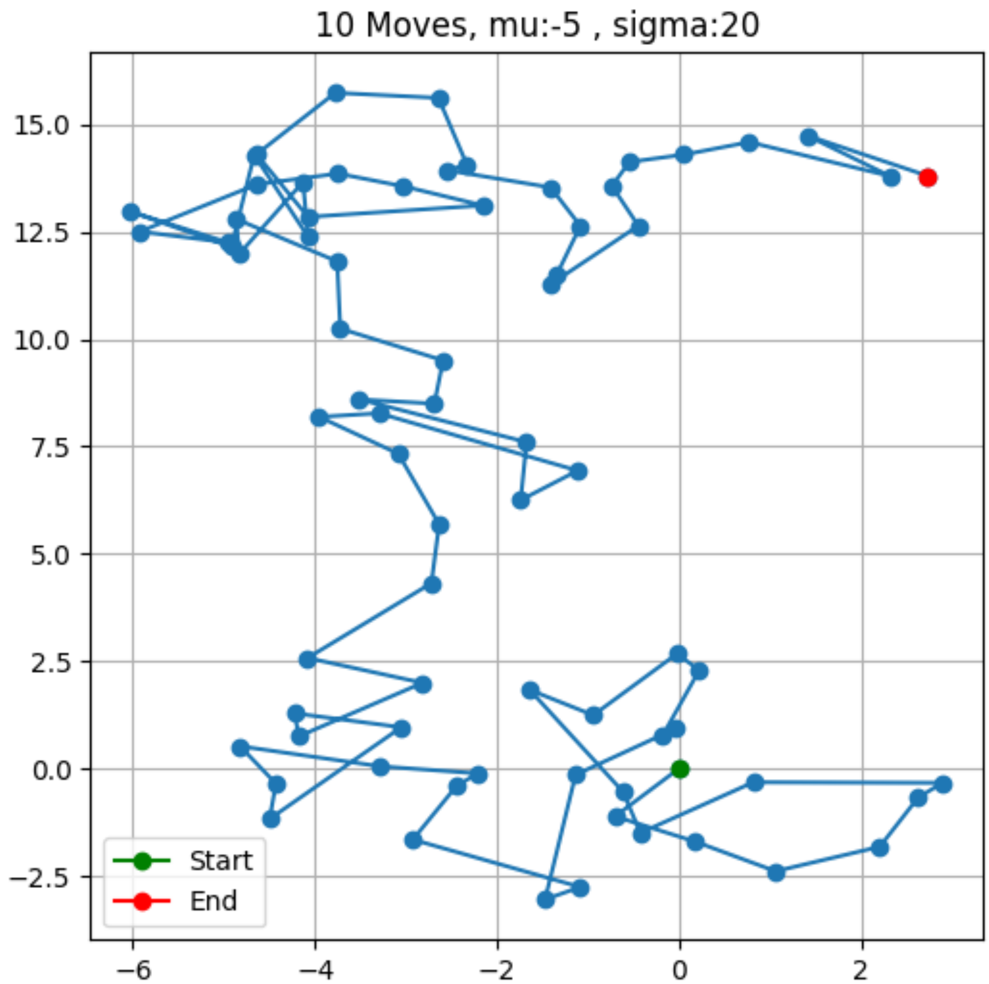

10 Moves, mu:-5 , sigma:10
Start
End
1
0
−1
−2
−3
−4
0
2
4
6

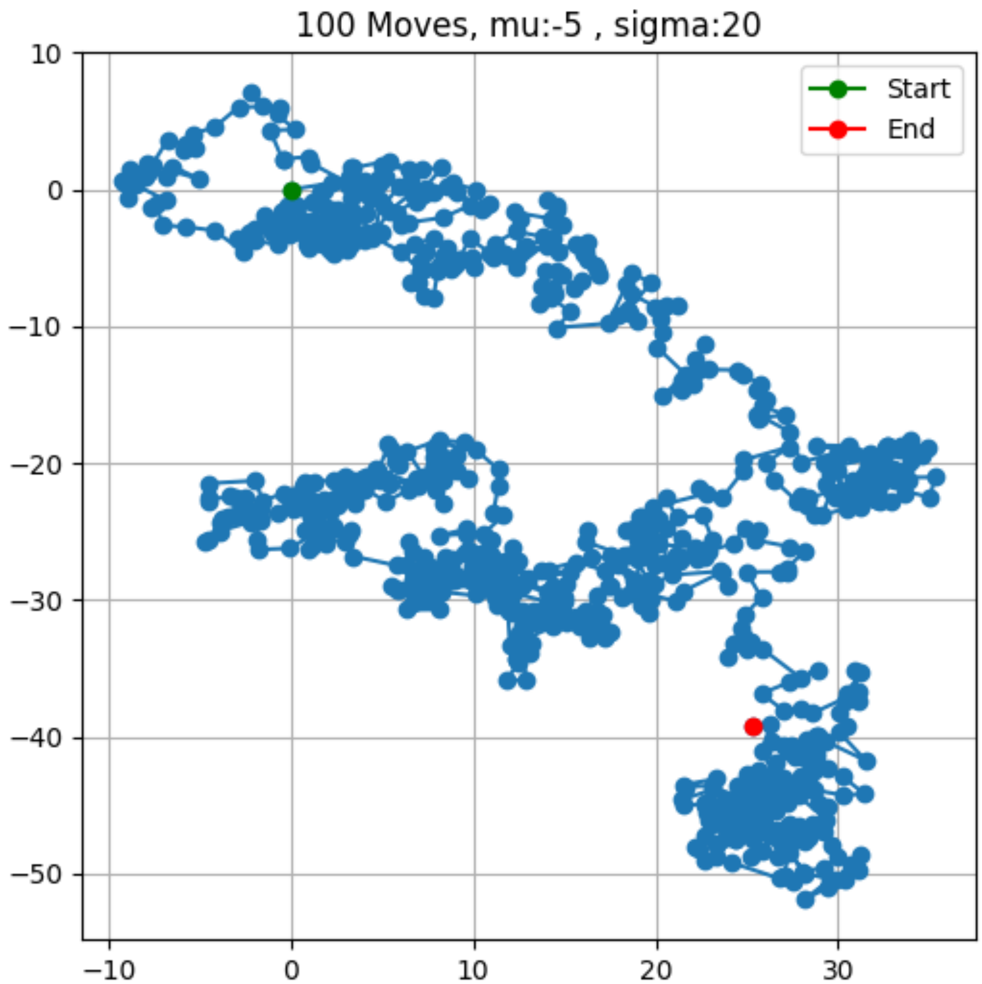

100 Moves, mu:-5 , sigma:10
Start
End
5
0
−5
−10
−15
−20
−25
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0

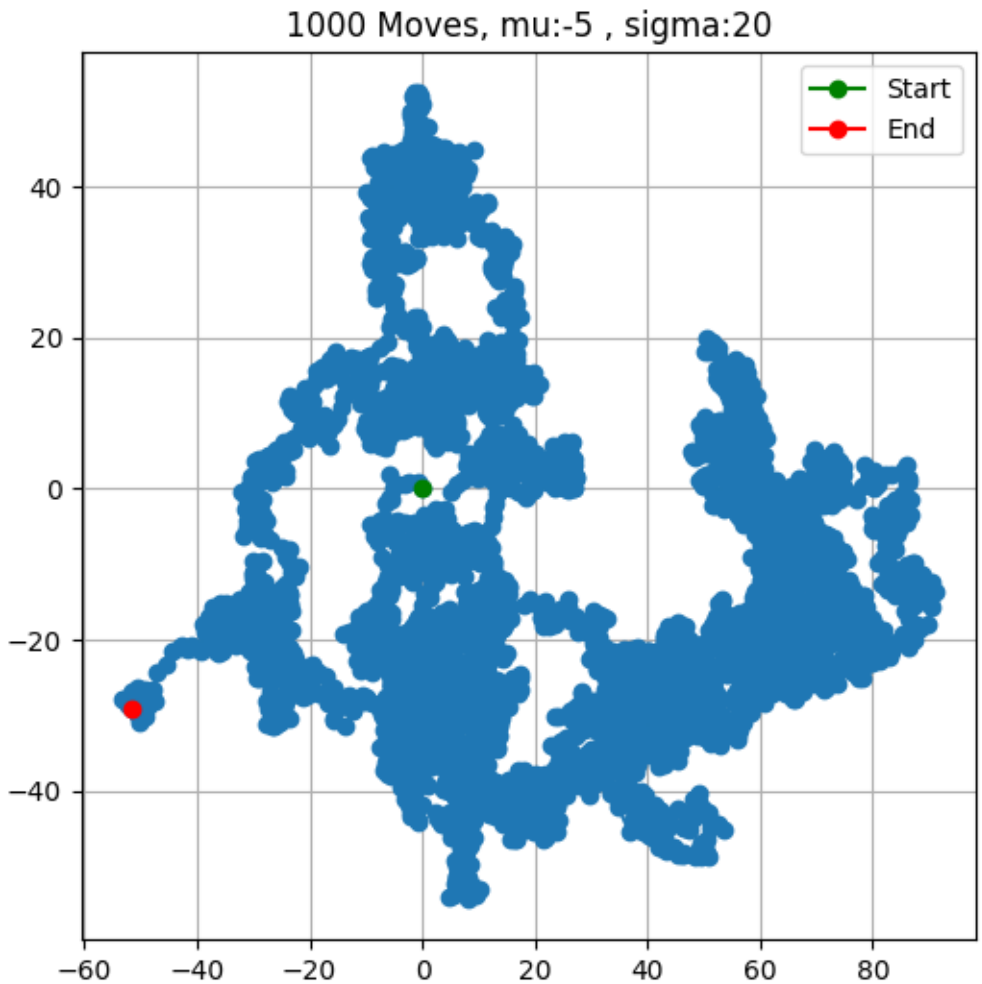

1000 Moves, mu:-5 , sigma:10
Start
End
20
10
0
−10
−20
−30
−50
−40
−30
−20
−10
0
10

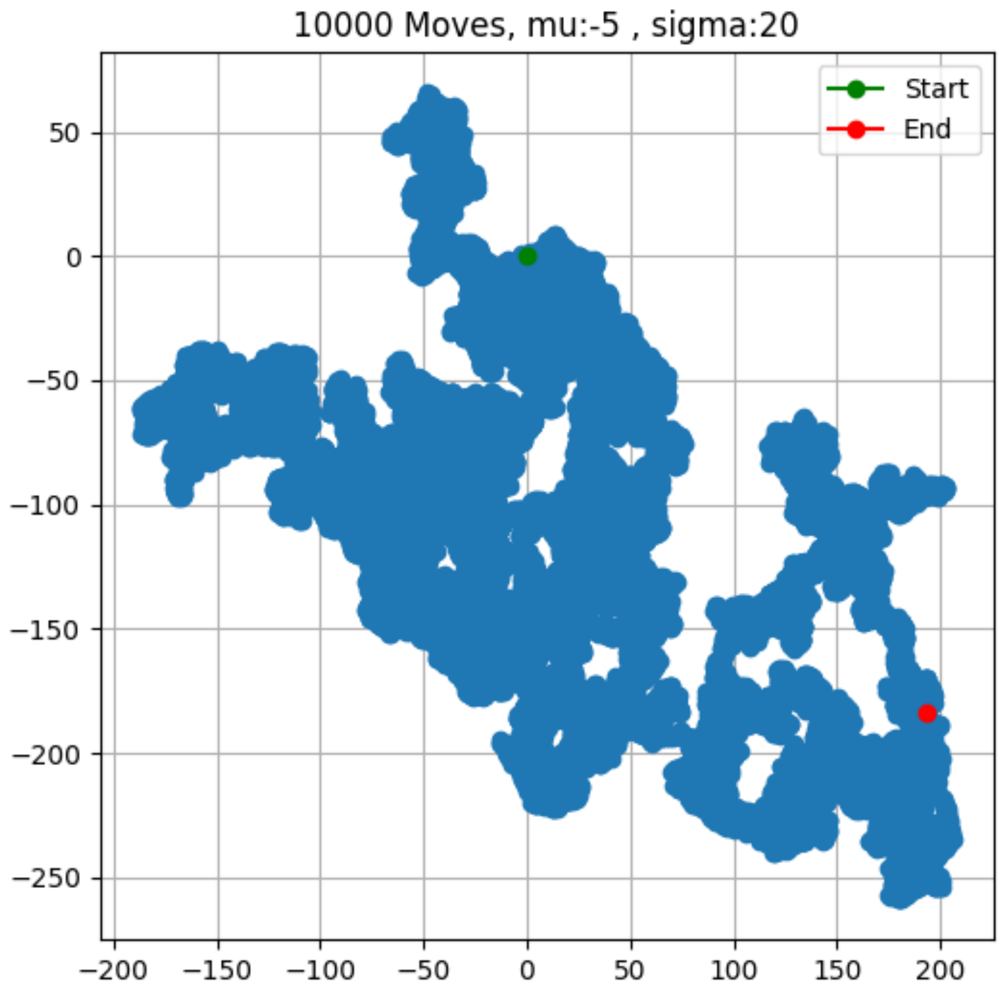

10000 Moves, mu:-5 , sigma:10
Start
End
150
100
50
0
−50
0
100
200
300
400

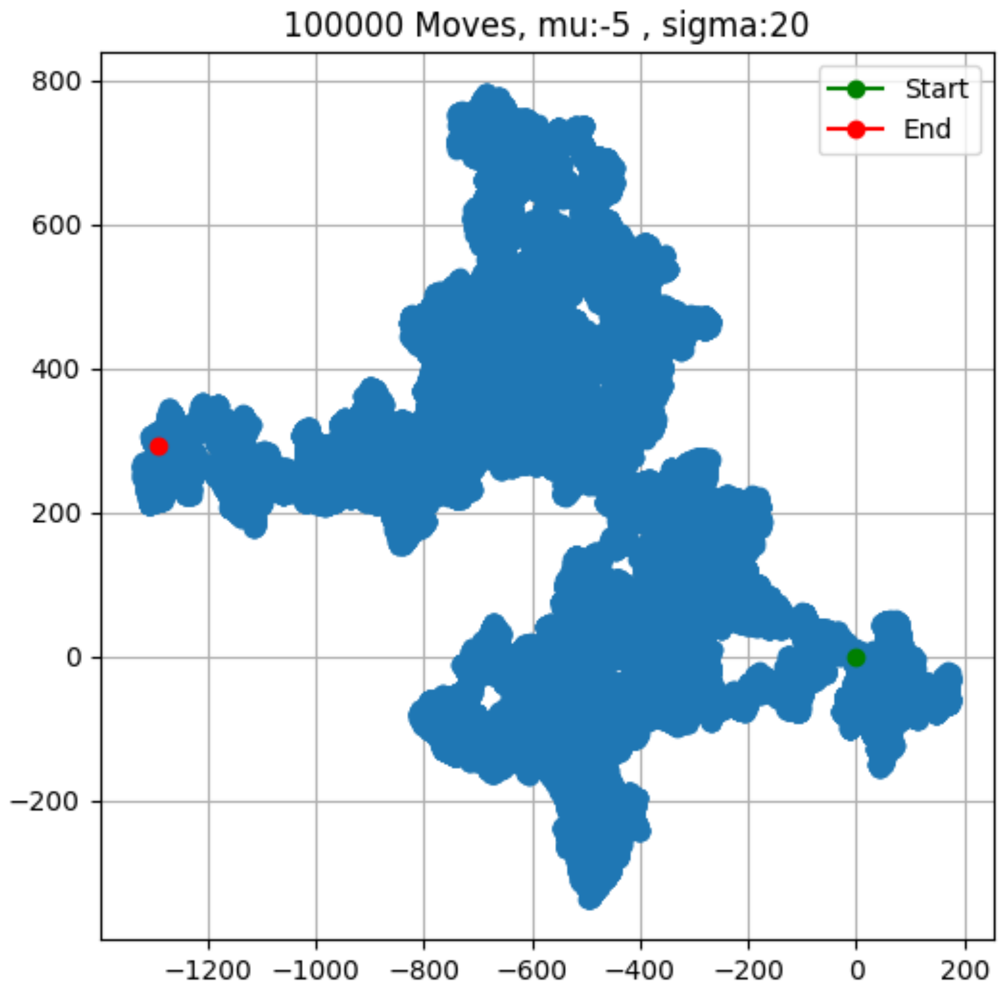

100000 Moves, mu:-5 , sigma:10
Start
End
200
100
0
−100
−200
−300
−400
−500
−600
−500
−400
−300
−200
−100
0

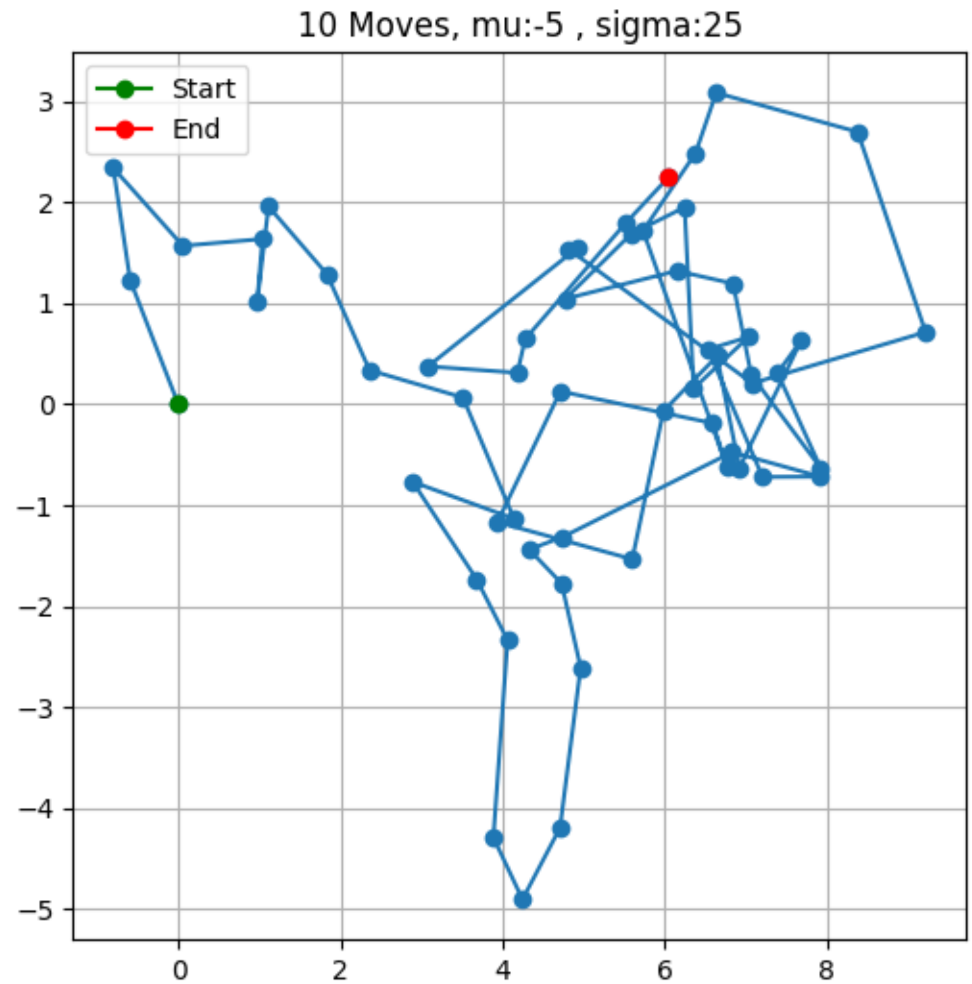

10 Moves, mu:-5 , sigma:20
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−6
−4
−2
0
2
Start
End

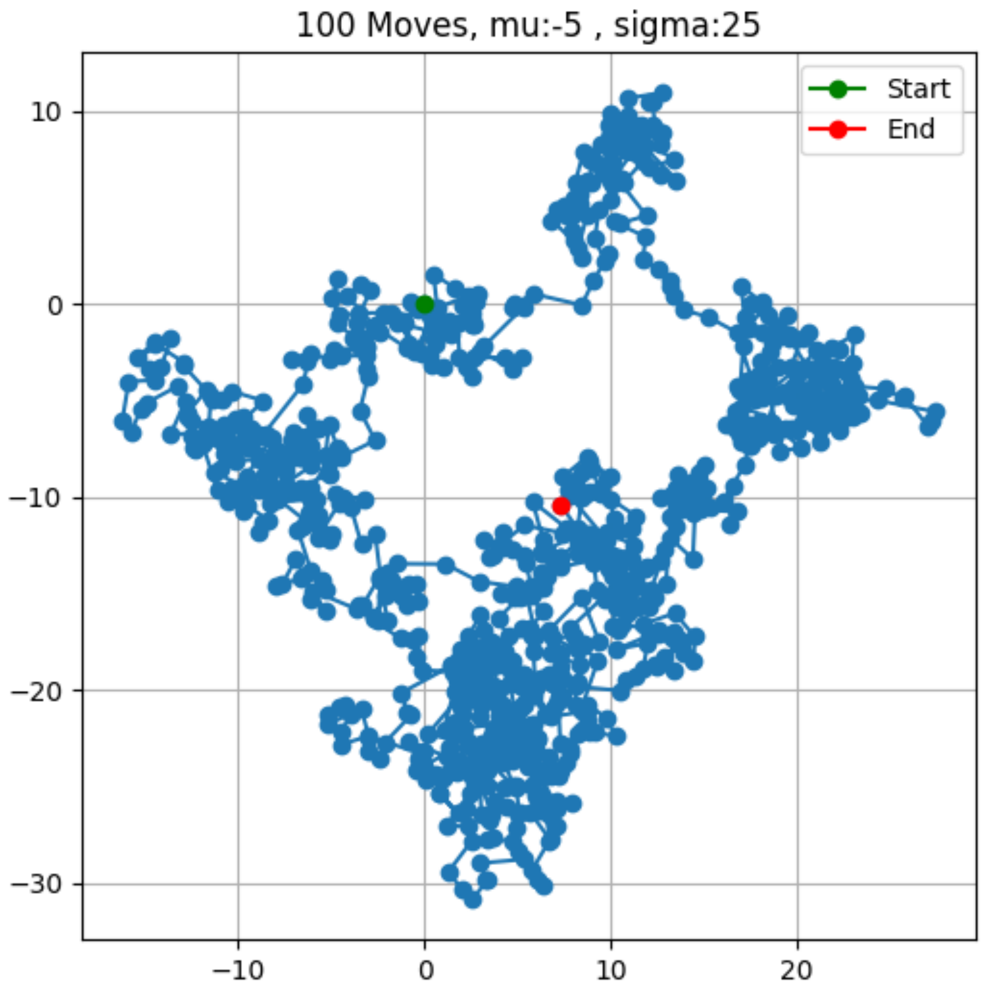

100 Moves, mu:-5 , sigma:20
Start
End
10
0
−10
−20
−30
−40
−50
−10
0
10
20
30


1000 Moves, mu:-5 , sigma:20
Start
End
40
20
0
−20
−40
−60
−40
−20
0
20
40
60
80

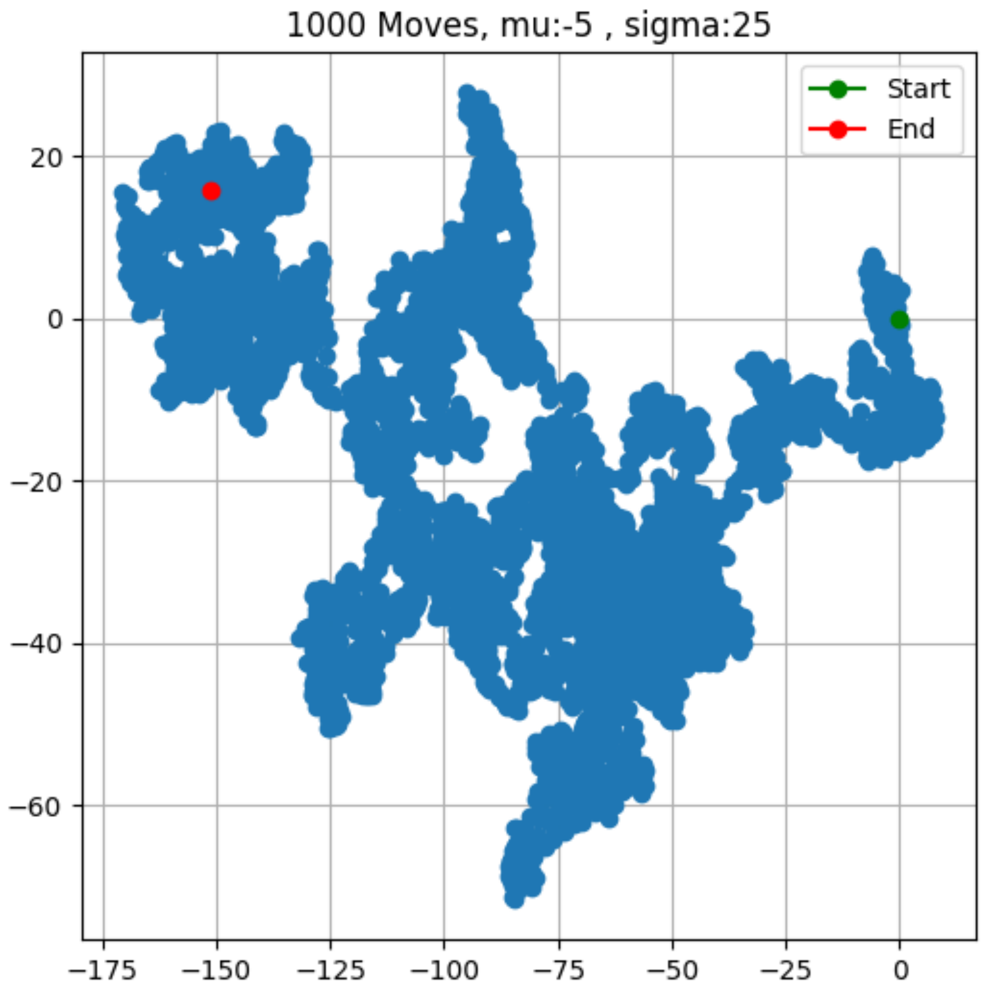

10000 Moves, mu:-5 , sigma:20
Start
End
50
0
−50
−100
−150
−200
−250
−200
−150
−100
−50
0
50
100
150
200

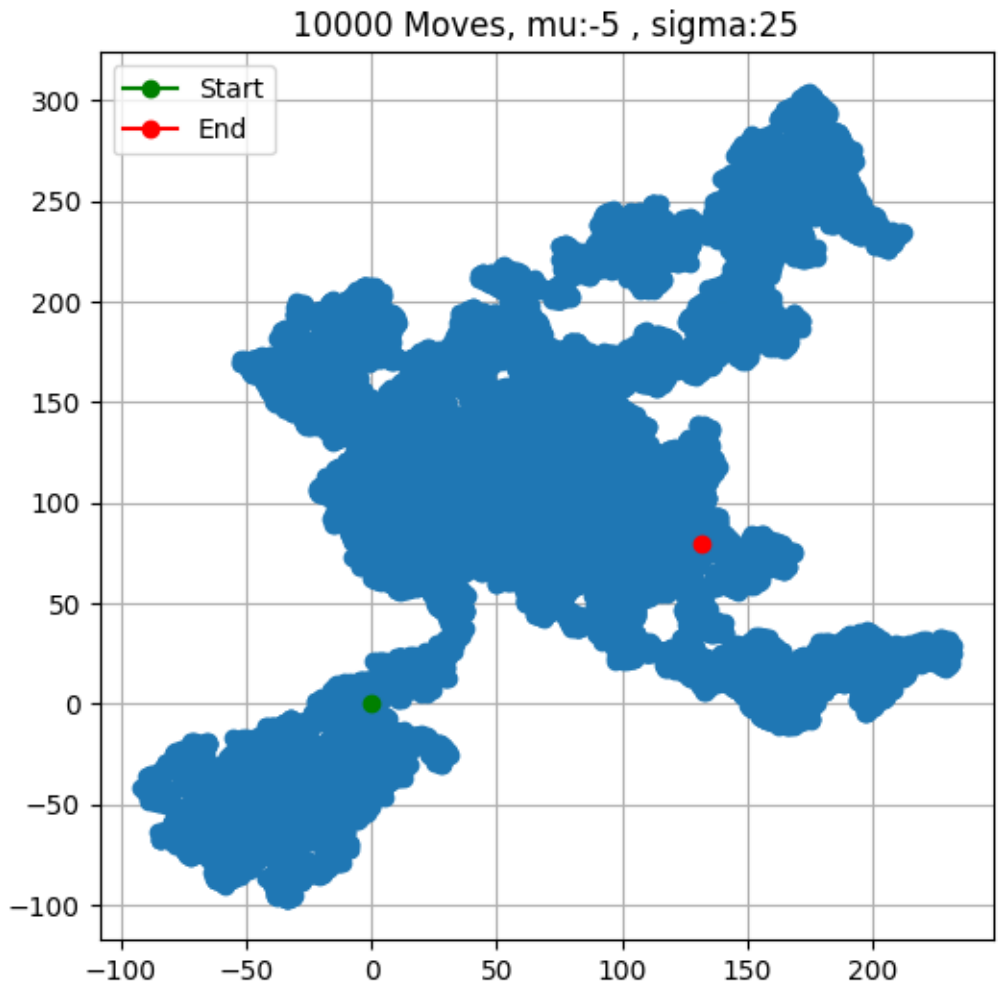

100000 Moves, mu:-5 , sigma:20
Start
End
800
600
400
200
0
−200
−1200
−1000
−800
−600
−400
−200
0
200

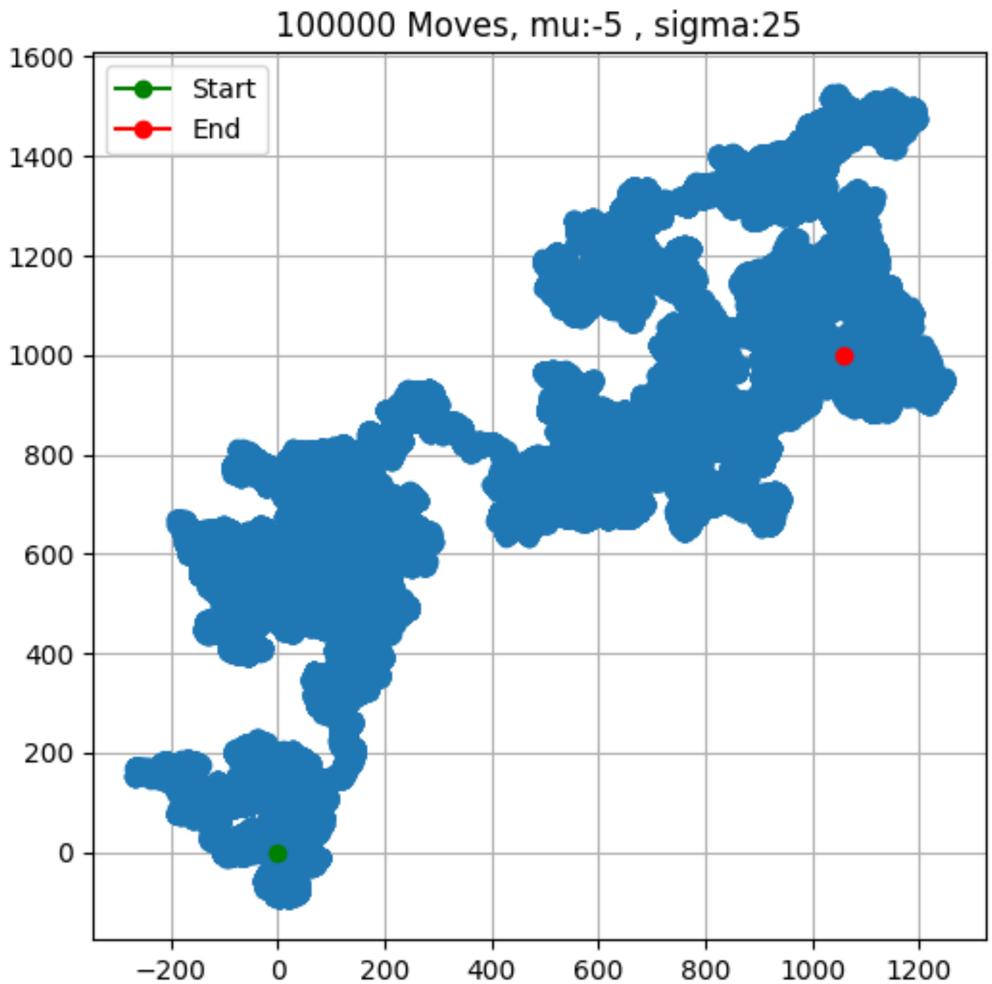

10 Moves, mu:-5 , sigma:25
Start
End
3
2
1
0
−1
−2
−3
−4
−5
0
2
4
6
8

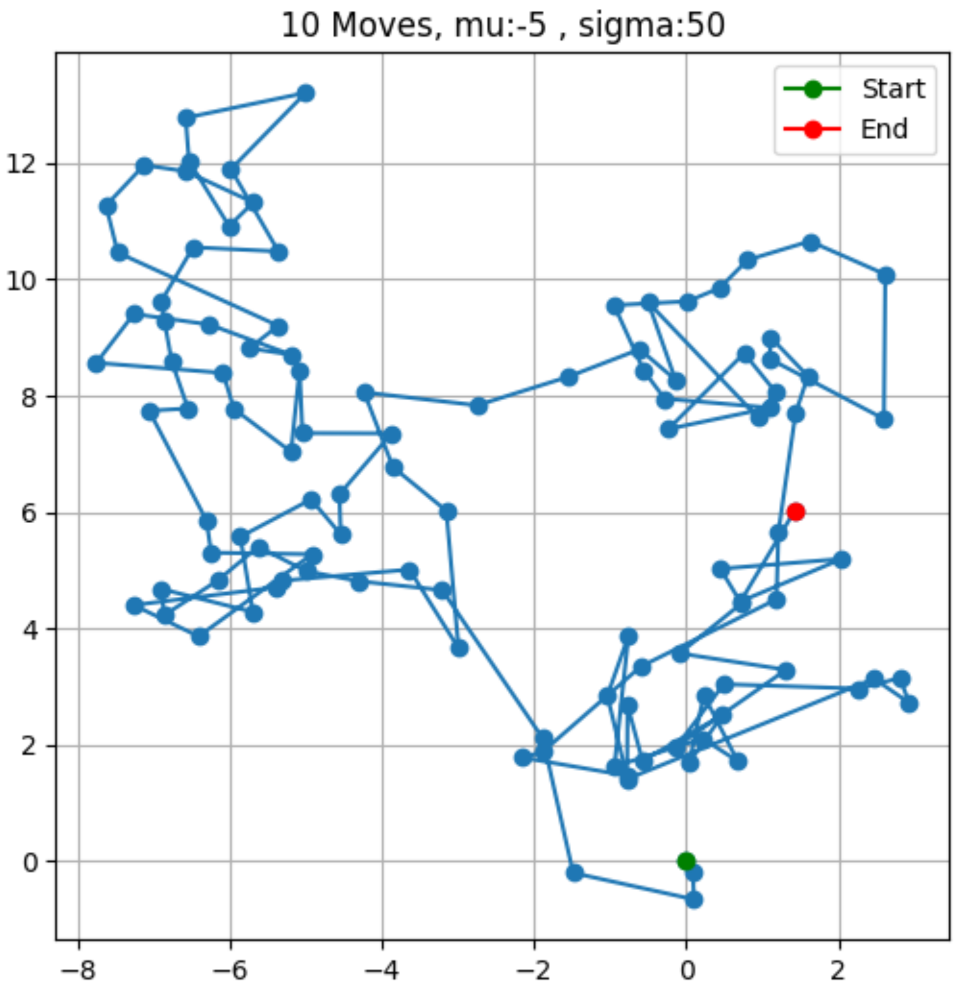


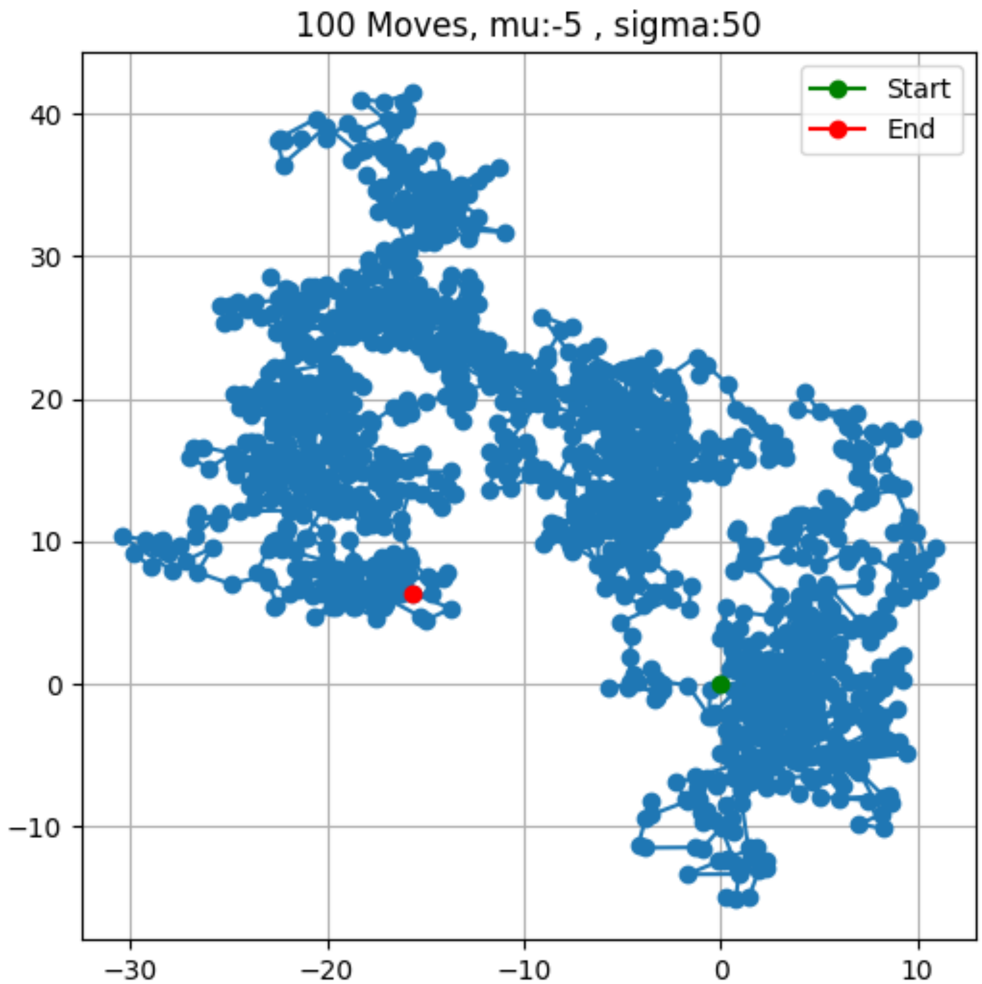
100 Moves, mu:-5 , sigma:25
Start
End
10
0
−10
−20
−30
−10
0
10
20

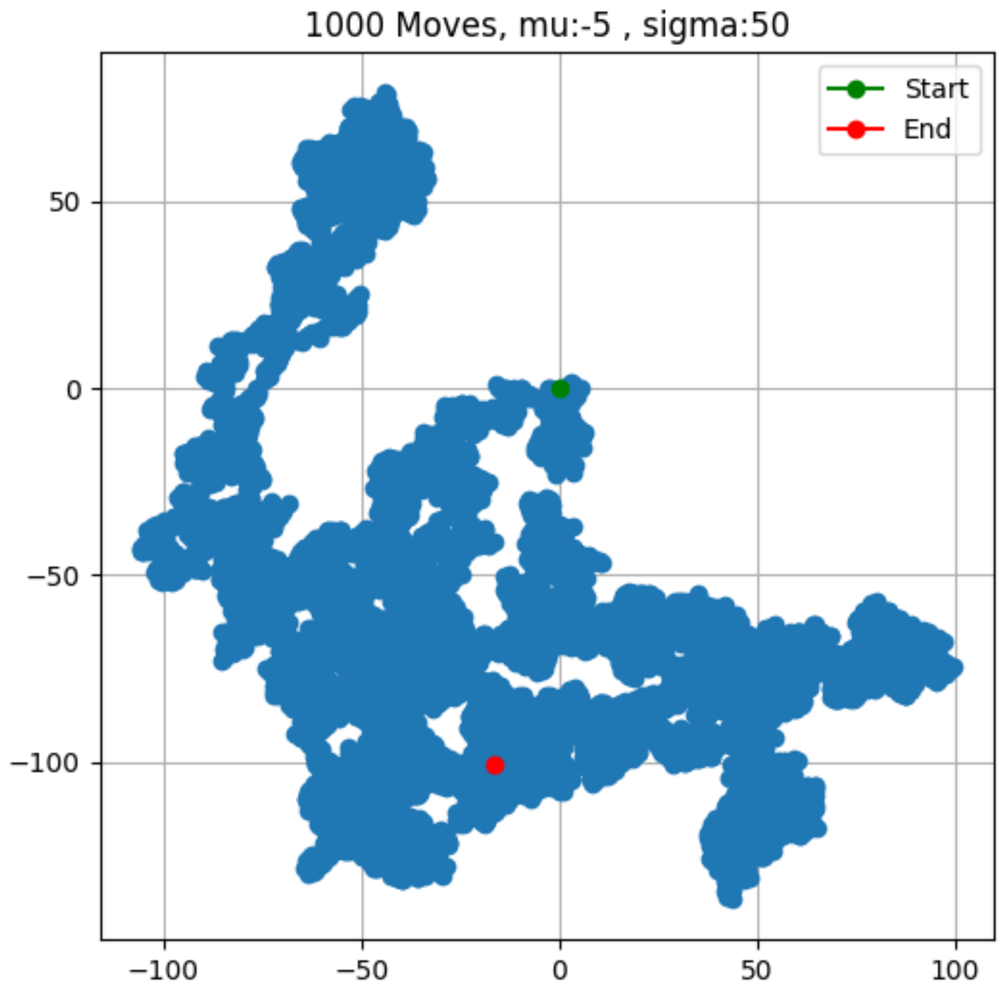

1000 Moves, mu:-5 , sigma:25
Start
End
20
0
−20
−40
−60
−175
−150
−125
−100
−75
−50
−25
0

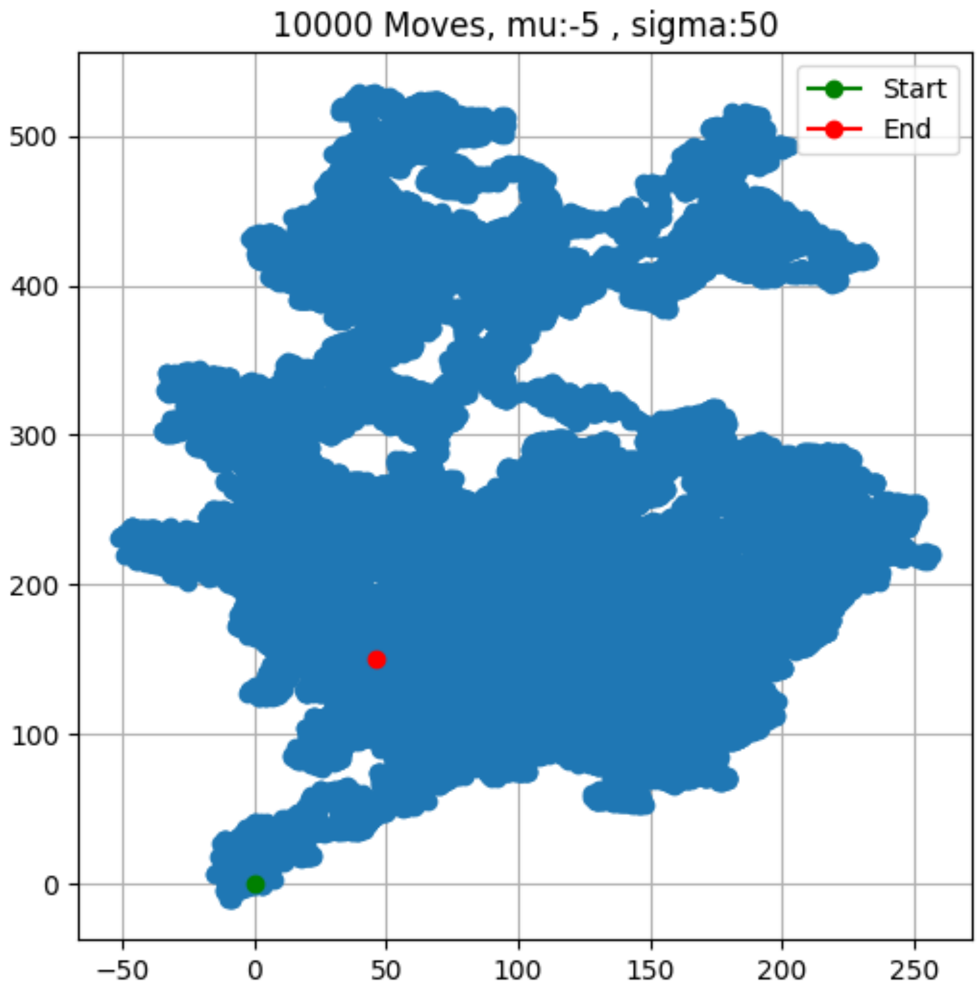

10000 Moves, mu:-5 , sigma:25
Start
End
300
250
200
150
100
50
0
−50
−100
−100
−50
0
50
100
150
200

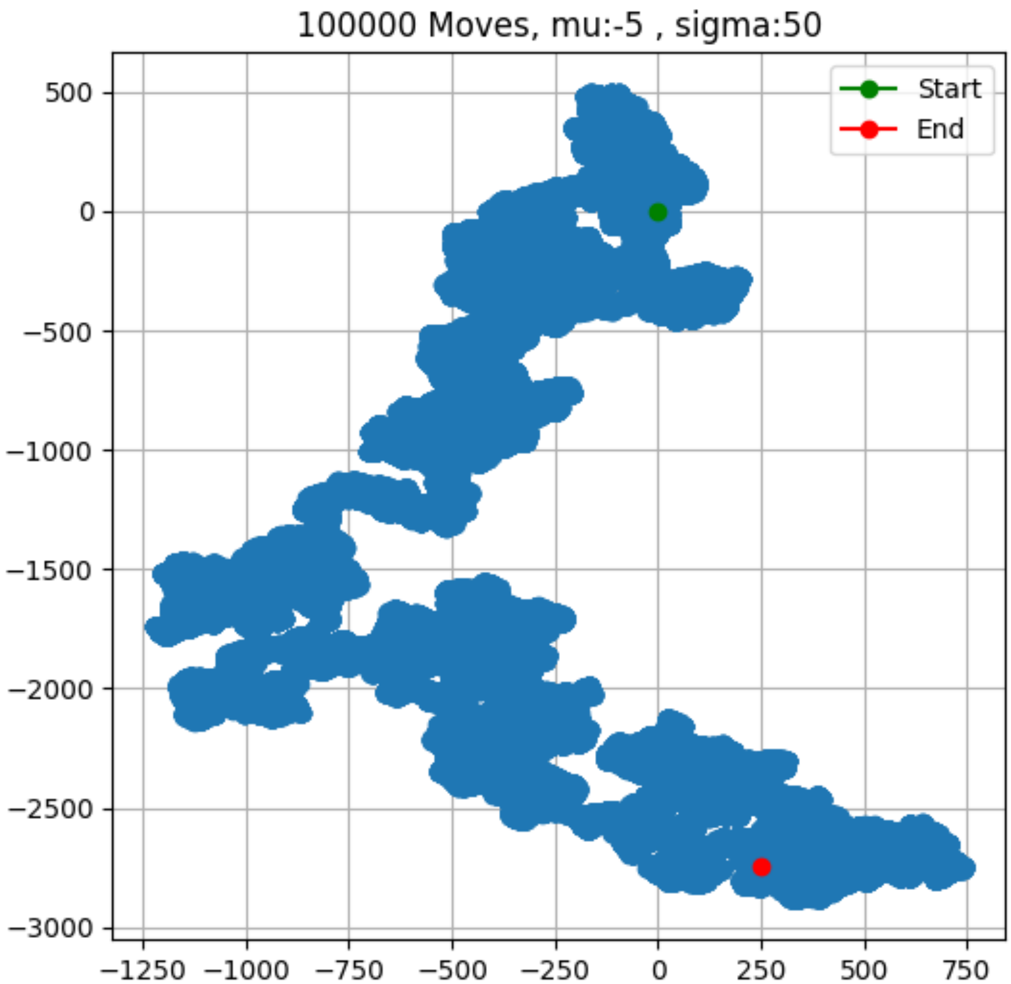

100000 Moves, mu:-5 , sigma:25
Start
End
1600
1400
1200
1000
800
600
400
200
0
−200
0
200
400
600
800
1000
1200

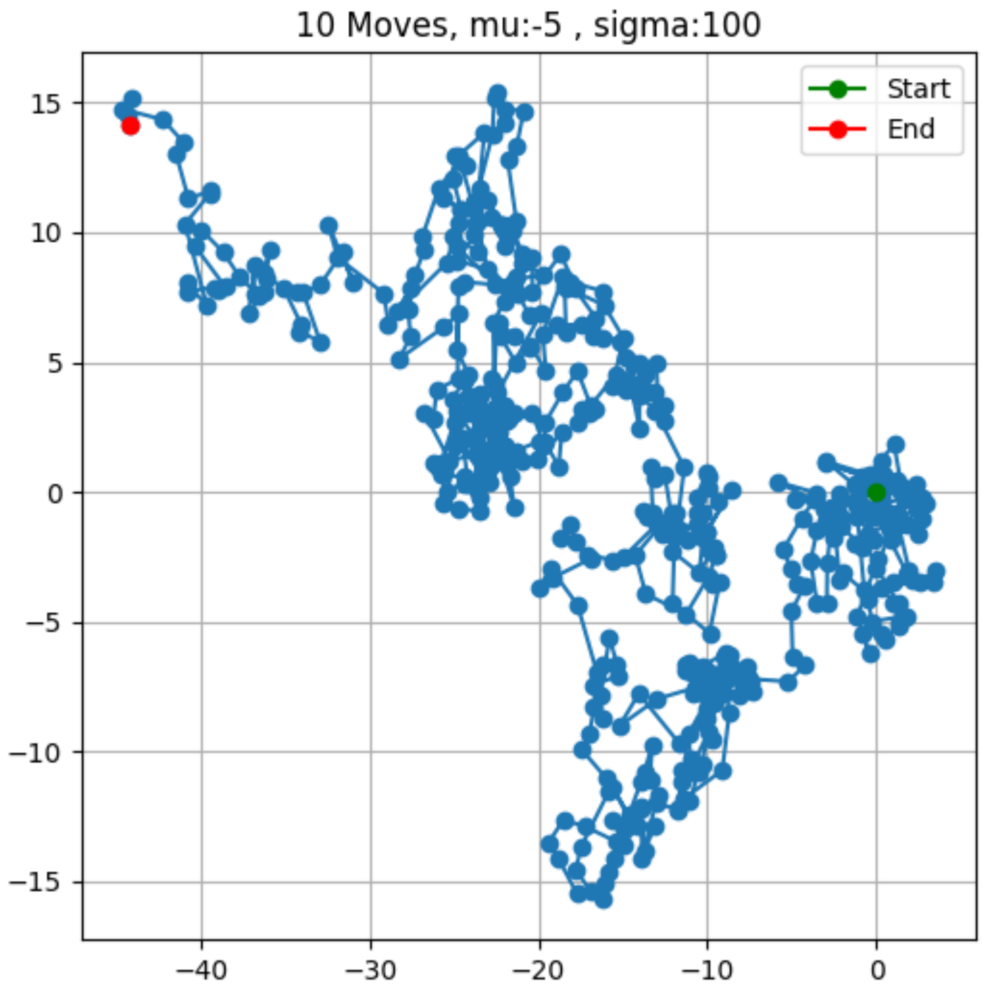

10 Moves, mu:-5 , sigma:50
Start
End
12
10
8
6
4
2
0
−8
−6
−4
−2
0
2


100 Moves, mu:-5 , sigma:50
Start
End
40
30
20
10
0
−10
−30
−20
−10
0
10

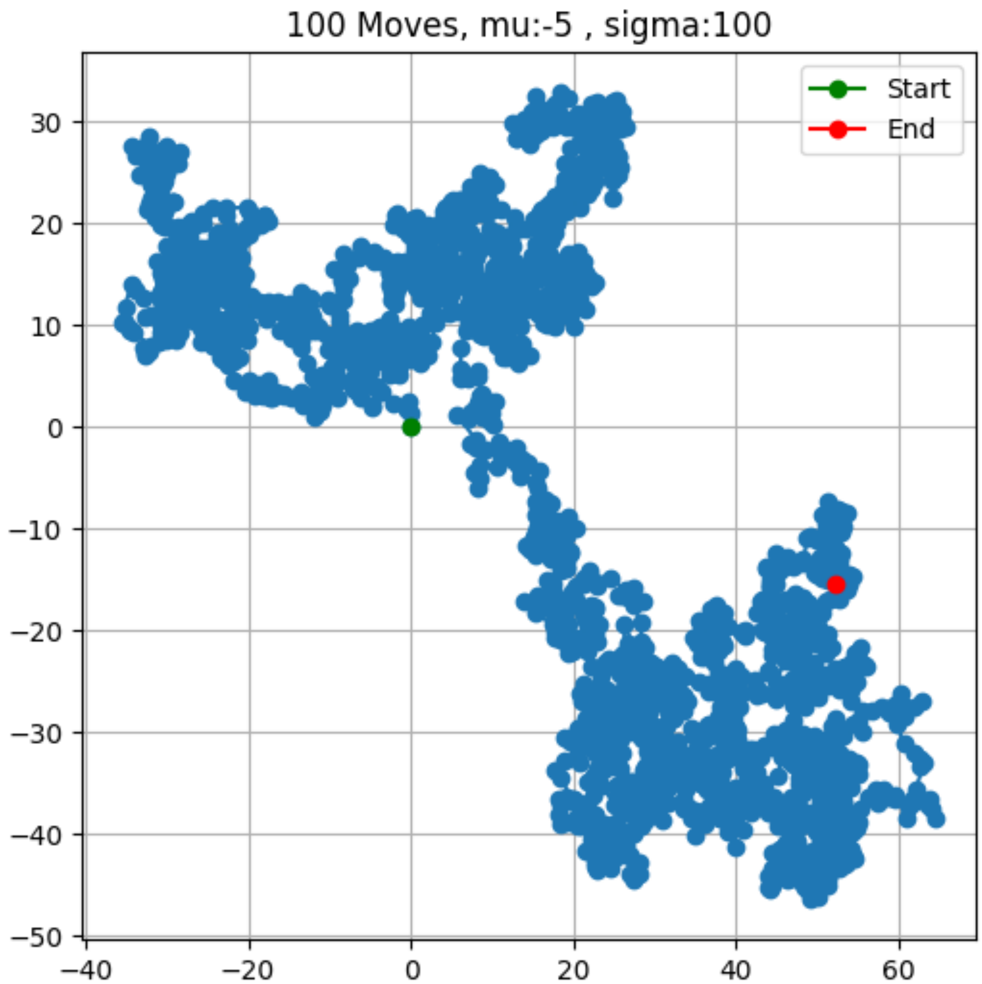

1000 Moves, mu:-5 , sigma:50
Start
End
50
0
−50
−100
−100
−50
0
50
100


10000 Moves, mu:-5 , sigma:50
Start
End
500
400
300
200
100
0
−50
0
50
100
150
200
250

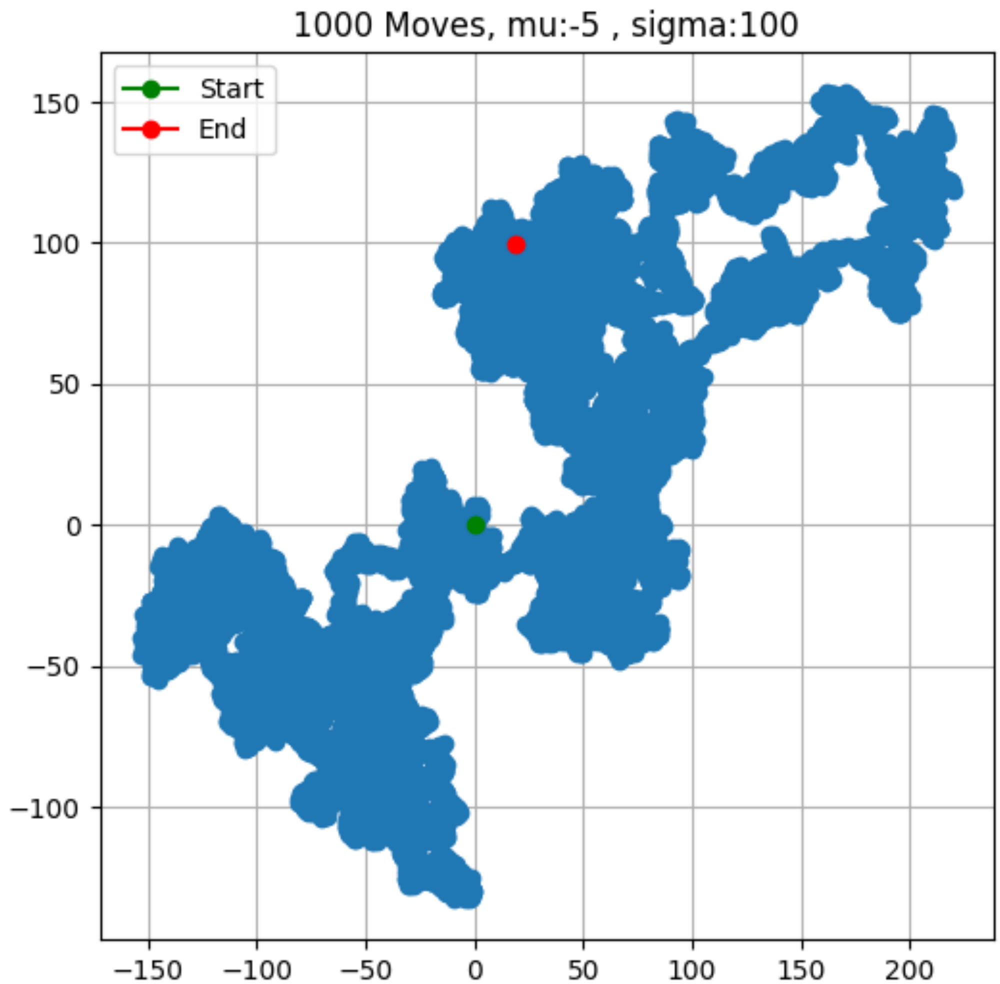

100000 Moves, mu:-5 , sigma:50
Start
End
500
0
−500
−1000
−1500
−2000
−2500
−3000
−1250
−1000
−750
−500
−250
0
250
500
750

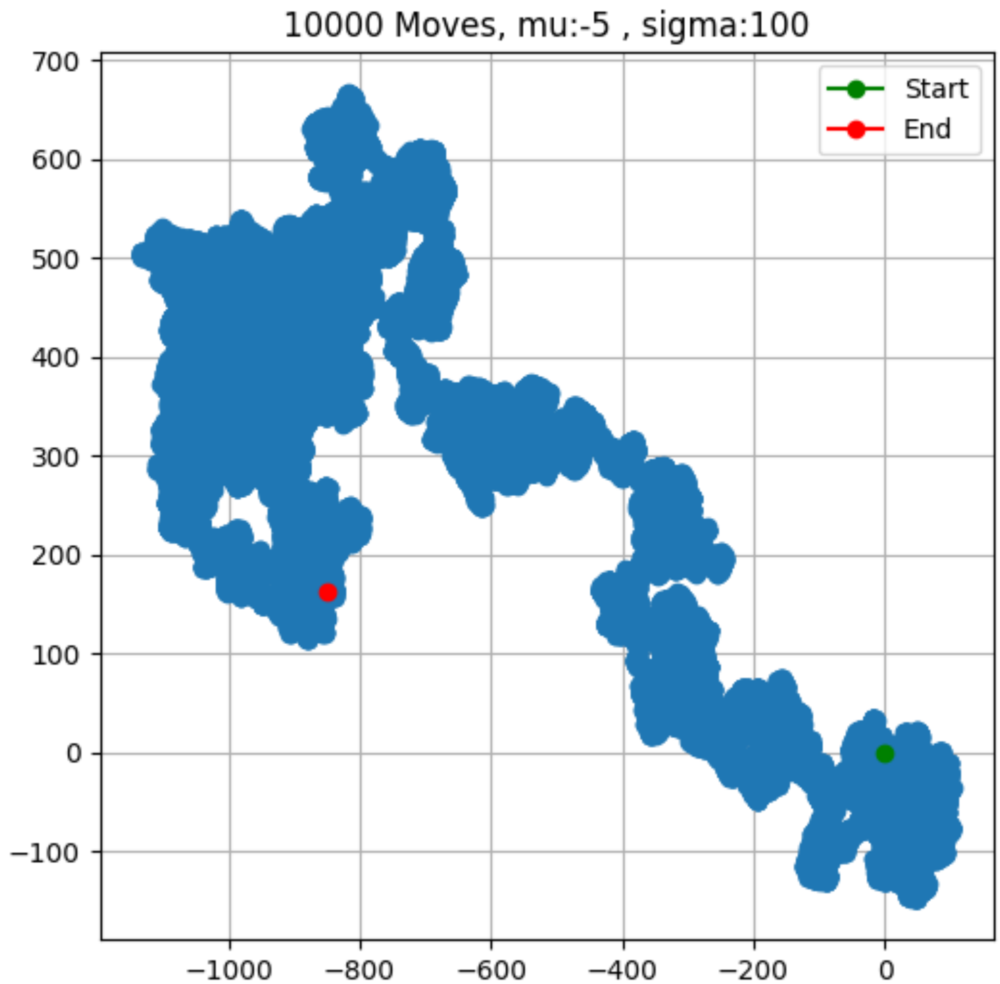


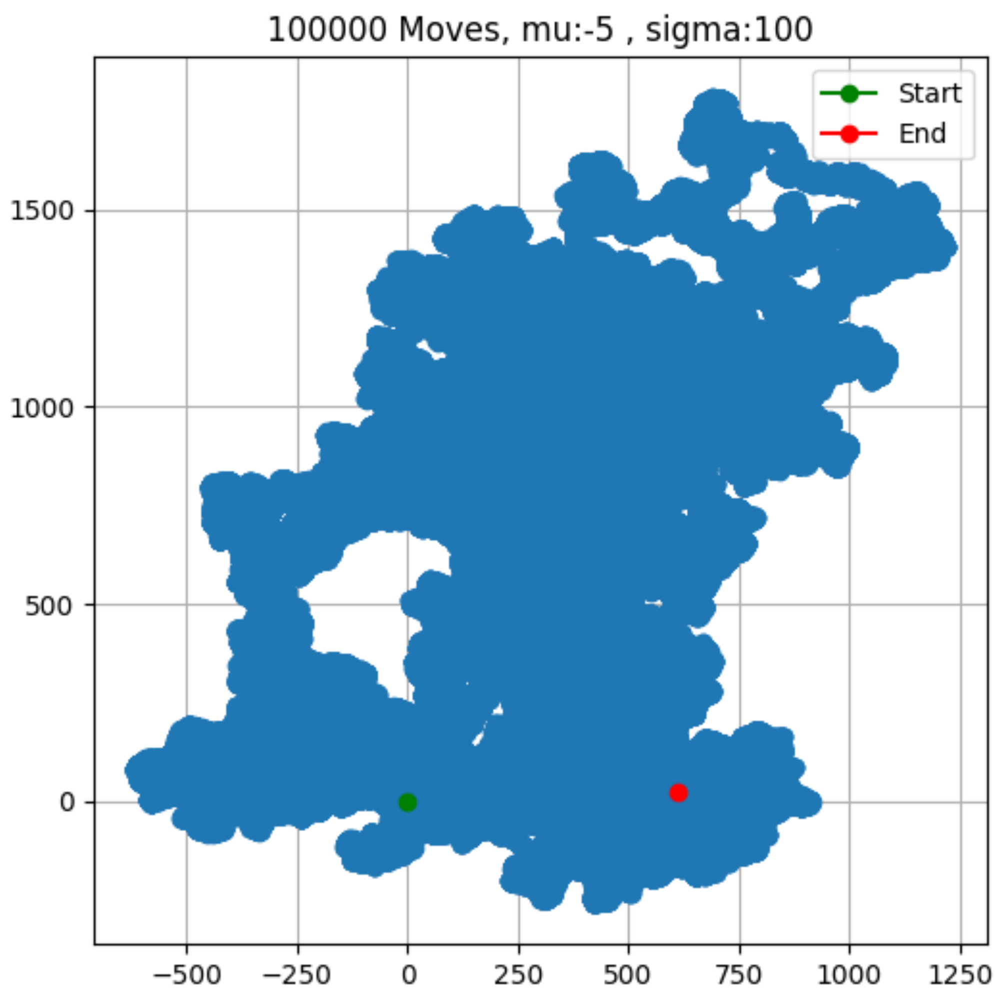
10 Moves, mu:-5 , sigma:100
Start
End
15
10
5
0
−5
−10
−15
−40
−30
−20
−10
0



100 Moves, mu:-5 , sigma:100
Start
End
30
20
10
0
−10
−20
−30
−40
−50
−40
−20
0
20
40
60


1000 Moves, mu:-5 , sigma:100
Start
End
150
100
50
0
−50
−100
−150
−100
−50
0
50
100
150
200


10000 Moves, mu:-5 , sigma:100
Start
End
700
600
500
400
300
200
100
0
−100
−1000
−800
−600
−400
−200
0



Question 16

```
[24]: def random_walk_2D_back_to_origin(n, dxdy):
          pos = [0, 0]
          for i in range(n):
              pos += dxdy[np.random.choice(len(dxdy), p=[1/len(dxdy) for i in␣
       ↪range(len(dxdy))])]
              if pos[0] == 0 and pos[1] == 0:
                  return True
          return False
      N = 1000
      n = 1000
      count = 0
      dxdy = np.array([[1, 0], [-1, 0], [0, 1], [0, -1]])
      for i in range(N):
          if random_walk_2D_back_to_origin(n, dxdy):
```

```
        count += 1
print(f'The probability of getting back to origin is {count/N}')
```

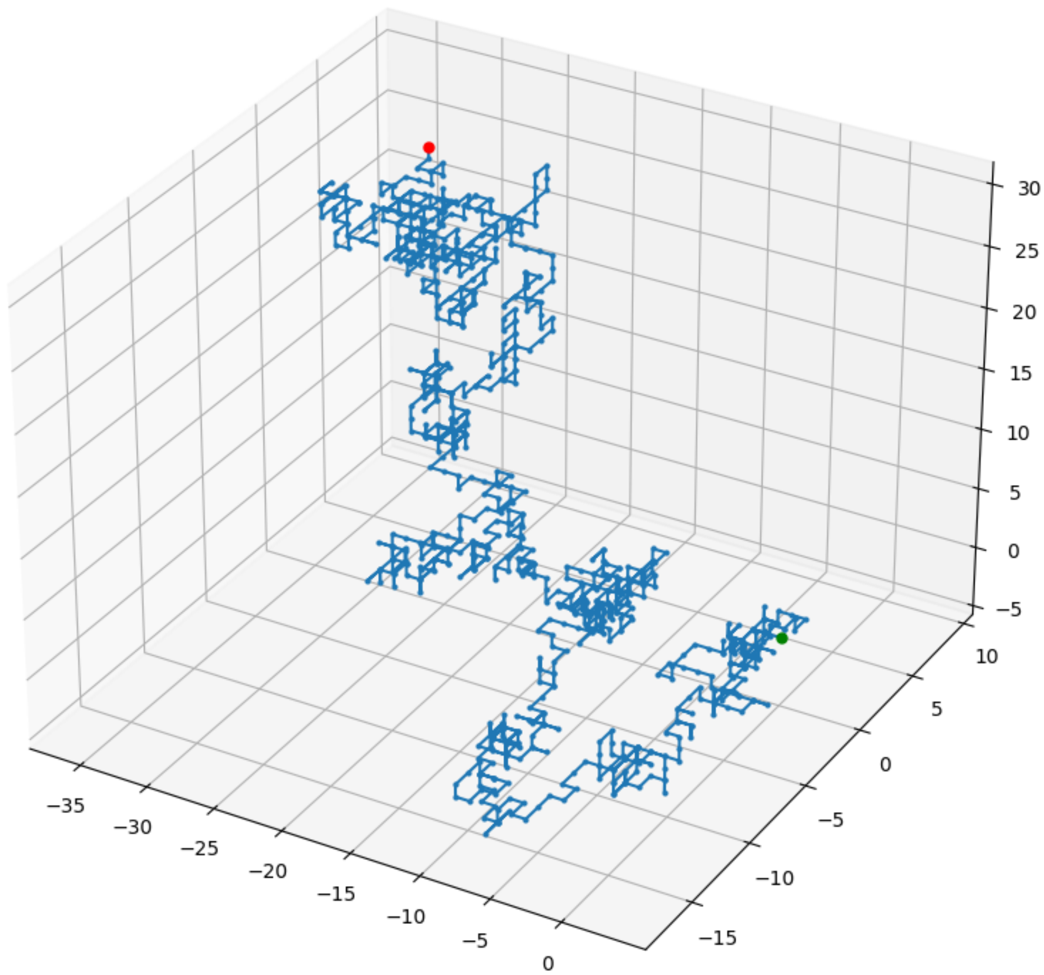
```
The probability of getting back to origin is 0.679
```

[25]:
```
count = 0
dxdy = np.array([[1, 1], [-1, -1], [-1, 1], [1, -1]])
for i in range(N):
    if random_walk_2D_back_to_origin(n, dxdy):
        count += 1
print(f'The probability of getting back to origin is {count/N}')
```

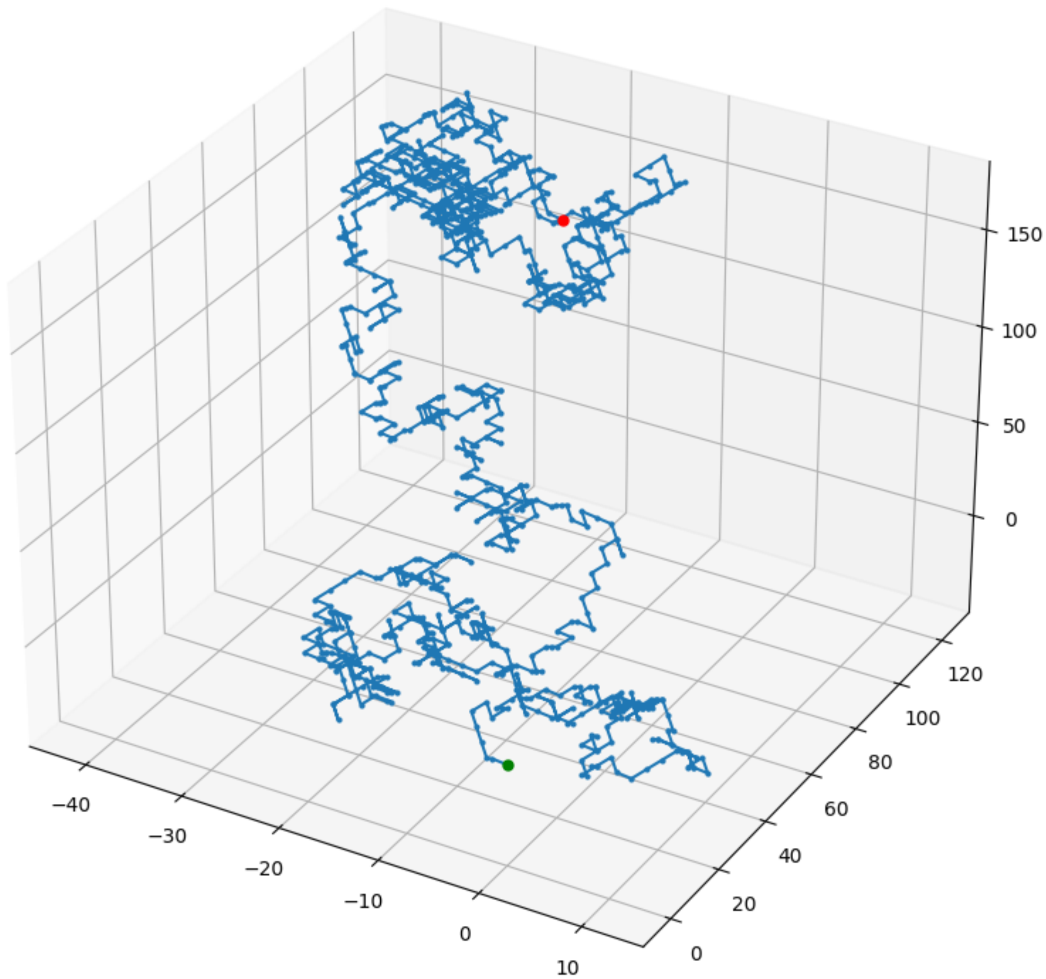
```
The probability of getting back to origin is 0.668
```

Question 17

[34]:
```
def random_walk_3D(n, dxdydz):
    pos = np.zeros((n+1, 3))
    pos[0] = [0, 0, 0]

    for i in range(1, n+1):
        x = dxdydz[np.random.choice(len(dxdydz), p=[1/len(dxdydz) for i in␣
 ↪range(len(dxdydz))])]
        pos[i] = pos[i-1] + x

    fig = plt.figure(figsize=(10, 10))
    ax = fig.add_subplot(111, projection='3d')
    ax.plot(pos[:, 0], pos[:, 1], pos[:, 2], linestyle='-', marker='o',␣
 ↪markersize=2)
    ax.plot(pos[0][0], pos[0][1], pos[0][2], marker='o', markersize=5,␣
 ↪color='green')
    ax.plot(pos[-1][0], pos[-1][1], pos[-1][2], marker='o', markersize=5,␣
 ↪color='red')

n = 1000
dxdydz = np.array([[1, 0, 0], [-1, 0, 0], [0, 1, 0], [0, -1, 0], [0, 0, 1], [0,␣
 ↪0, -1]])
random_walk_3D(n ,dxdydz)
```

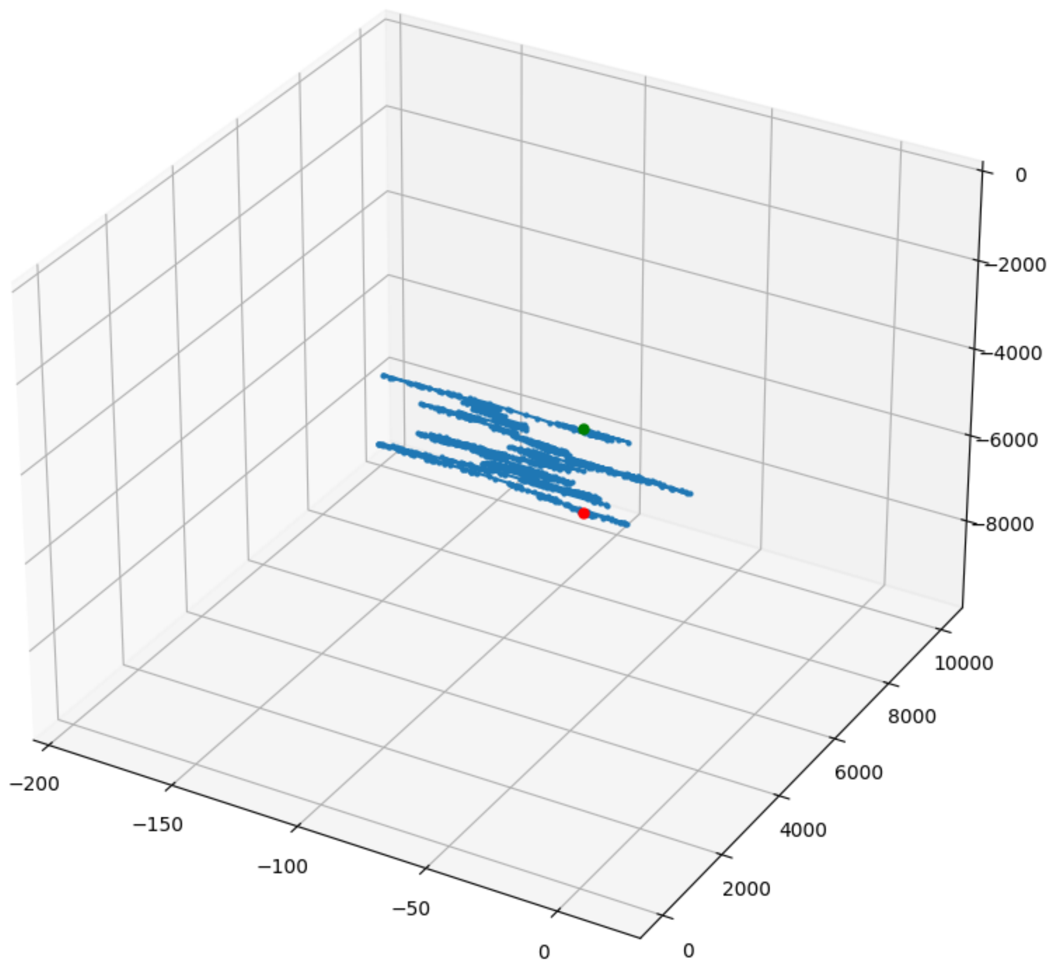


[35]:
```
dxdydz = np.array([[1, 2, 4], [-1, 2, 4], [1, -2, 4], [1, 2, -4], [1, -2, -4],
↪[-1, 2, -4], [-1, -2, 4], [-1, -2, -4]])
random_walk_3D(n ,dxdydz)
```



[36]:
```
def random_walk_3D_arbitrary_distribution(n):
    pos = np.zeros((n+1, 3))
    pos[0] = [0, 0, 0]

    for i in range(1, n+1):
        x1 = np.random.normal(0, 4)
        x2 = np.random.gamma(2, 5)
        x3 = np.random.laplace(-10, 3)
        pos[i] = pos[i-1] + [x1, x2, x3]

    fig = plt.figure(figsize=(10, 10))
    ax = fig.add_subplot(111, projection='3d')
```

```
    ax.plot(pos[:, 0], pos[:, 1], pos[:, 2], linestyle='-', marker='o',␣
↪markersize=2)
    ax.plot(pos[0][0], pos[0][1], pos[0][2], marker='o', markersize=5,␣
↪color='green')
    ax.plot(pos[-1][0], pos[-1][1], pos[-1][2], marker='o', markersize=5,␣
↪color='red')

n = 1000
random_walk_3D_arbitrary_distribution(n)
```



Question 18

[37]:
```python
def random_walk_3D_back_to_origin(n ,dxdydz):
    pos = [0, 0, 0]

    for i in range(n):
        pos += dxdydz[np.random.choice(len(dxdydz), p=[1/len(dxdydz) for i in␣
↪range(len(dxdydz))])]
        if pos[0] == 0 and pos[1] == 0 and pos[-1] == 0:
            return True
    return False

N = 1000
n = 100
count = 0
dxdydz = np.array([[1, 1, 1], [1, 1, -1], [1, -1, 1], [-1, 1, 1], [1, -1, -1],␣
↪[-1, -1, 1], [-1, 1, -1], [-1, -1, -1]])
for i in range(N):
    if random_walk_3D_back_to_origin(n ,dxdydz):
        count += 1

print(f'The probability of getting back to origin is {count/N}')
```

```
The probability of getting back to origin is 0.278
```

Question 19

[ ]:
```python
#According to the results of Q16 and Q18, we can see that the probability of␣
↪returning to the origin
#in Q16 is much higher than Q18, becuase the degree of freedom in Q18 is higher␣
↪and therefor the probability of getting far
#from origin becomes higher than Q16
```

Question 20

[42]:
```python
from matplotlib.animation import FuncAnimation
def update(time):
    ax.clear()

    mu_x, mu_y = 0, 0
    var_x, var_y = 5 + 0.5 * np.cos(1 * time), 5 + 0.5 * np.cos(1 * time)

    x = np.linspace(-4, 4, 100)
    y = np.linspace(-4, 4, 100)

    X, Y = np.meshgrid(x, y)

    Z = (np.exp(-(X - mu_x)**2 / (2 * var_x)) * np.exp(-(Y - mu_y)**2 / (2 *␣
↪var_y)))
```

```
    ax.imshow(Z, cmap="rainbow", interpolation="nearest", origin="lower",␣
↪aspect="auto")

fig, ax = plt.subplots()
animation = FuncAnimation(fig, update, frames=100, interval=100, repeat=False)
```