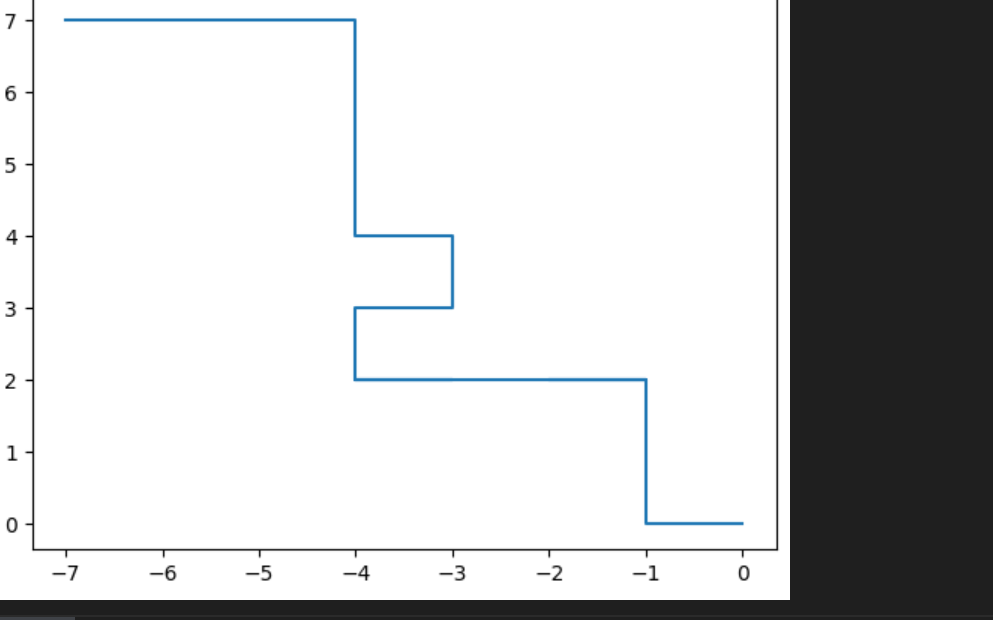
# **RANDOM WALK SIMULATION**

Random walk simulations model processes with random, step-by-step movements. This is often used to study systems with uncertainty. Examples include particle motion in physics, PageRank in web analysis, Monte Carlo algorithms in computation, and stock price modeling in finance. These walks can be simple (equal probabilities), biased (favouring certain directions), or multi-dimensional, helping to analyse and solve complex, probabilistic problems.

**GRAPH ONE:**



import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

prob = [0.98, 0.98, 0.32, 0.54, 0.54]

RN = [6, 2, 0, 87, 8, 596, 7, 7, 9, 87, 4, 8, 2, 6, 2, 1, 3, 93, 87, 44]

x = [0]

y = [0]

F = [0, 1, 2, 3, 4]

L = [5, 6, 7]

R = [8, 9] # Right movements

# Loop through the random numbers and simulate the random walk

for i in range(len(RN)):

if RN[i] in F:

x.append(x[i]) # No change in x (moving forward), just update y

y.append(y[i] + 1) # Move forward in the y-axis

elif RN[i] in R:

x.append(x[i] + 1) # Move right along the x-axis

y.append(y[i]) # Keep the y-coordinate the same

else:

x.append(x[i] -1) # Move left along the x-axis

*y.append(y[i])* # Keep the y-coordinate the same

# Store the results in a DataFrame for analysis

data = {"X Coordinate": x, "Y Coordinate": y}

df = pd.DataFrame(data)

# Output the data

print(df)

# Plot the random walk path

plt.figure(figsize=(8, 6))

plt.plot(x, y, marker='o', linestyle='-', color='b', label='Random Walk Path')

plt.title('Random Walk Simulation')

plt.xlabel('X Coordinate')

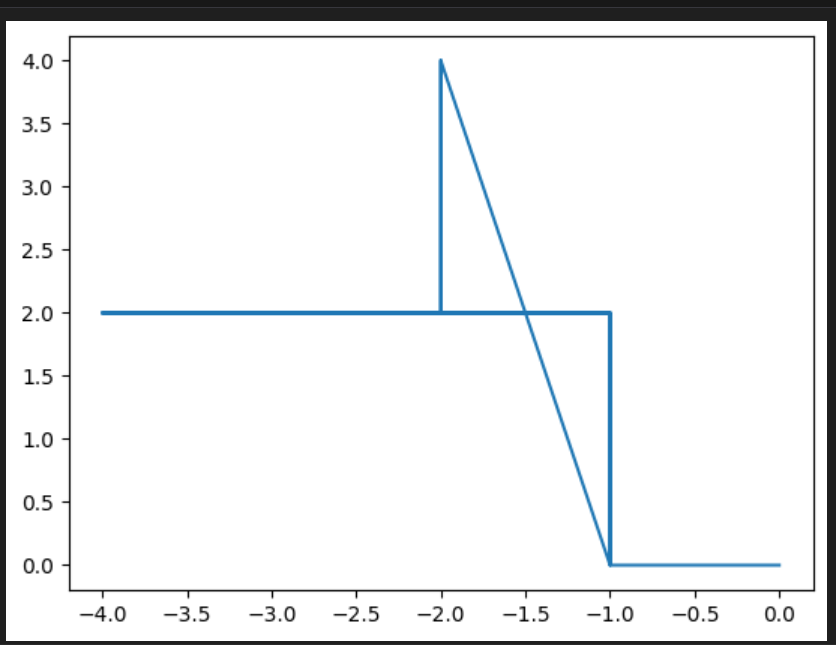
plt.ylabel('Y Coordinate')

plt.legend()

plt.grid(True)

plt.show()

**GRAPH 2:**



import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

# Define probabilities and random numbers

prob = [0.98, 0.98, 0.32, 0.54, 0.54] # Example probabilities

RN = [6, 2, 0, 87, 8, 596, 7, 7, 9, 87, 4, 8, 2, 6, 2, 1, 3, 93, 87, 44] # Example random numbers

# Initialize coordinates

x = [0]

y = [0]

# Movement rules

F = [0, 1, 2, 3, 4] # Forward

L = [5, 6, 7] # Left

R = [8, 9] # Right

# Simulate random walk

for i in range(len(RN)):

if RN[i] in F:

x.append(x[i])

y.append(y[i] + 1) # Move forward

elif RN[i] in R:

x.append(x[i] + 1) # Move right

y.append(y[i])

else:

x.append(x[i] - 1) # Move left

y.append(y[i])

# Store the results in a DataFrame

data = {"X Coordinate": x, "Y Coordinate": y}

df = pd.DataFrame(data)

# Display the DataFrame

print(df)

# Plot the random walk

plt.figure(figsize=(8, 6))

plt.plot(x, y, marker='o', linestyle='-', color='b', label='Random Walk Path')

plt.title('Random Walk Simulation')

plt.xlabel('X Coordinate')

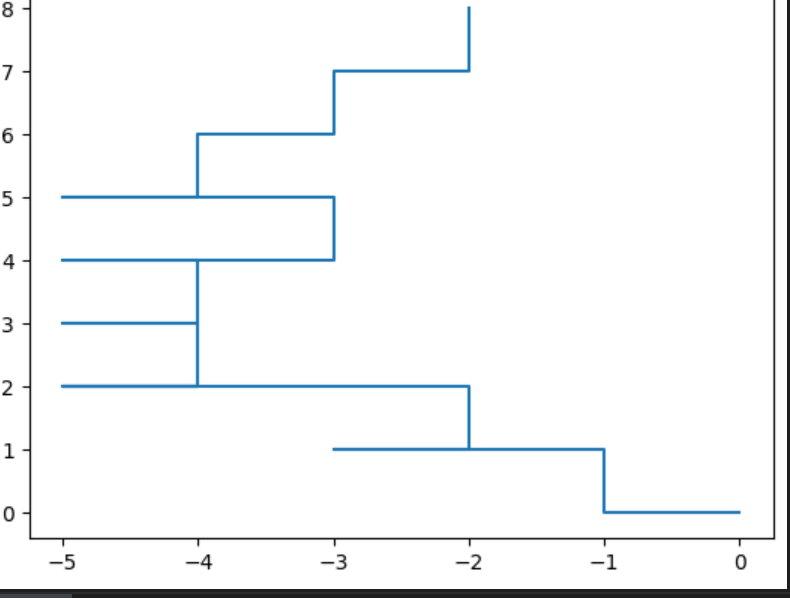
plt.ylabel('Y Coordinate')

plt.legend()

plt.grid()

plt.show()

**GRAPH 3:**



import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

prob = [0.50, 0.20 ,0.30]

RN = [6, 2,10, 6, 8, 3, 7, 7, 11, 8, 11, 8, 2, 6, 8, 1, 7, 9, 8, 4, 7, 32, 9, 4, 8, 3, 8, 1]

x=[0]

y=[0

L = [5, 6, 7]

R = F = [0, 1, 2, 3, 4]

[8, 9]

for i in range(len(RN)):

    if RN[i] in F:

        x.extend([x[i]])

        y.extend([y[i]+1])

    elif RN[i] in R:

        x.extend([x[i]+1])

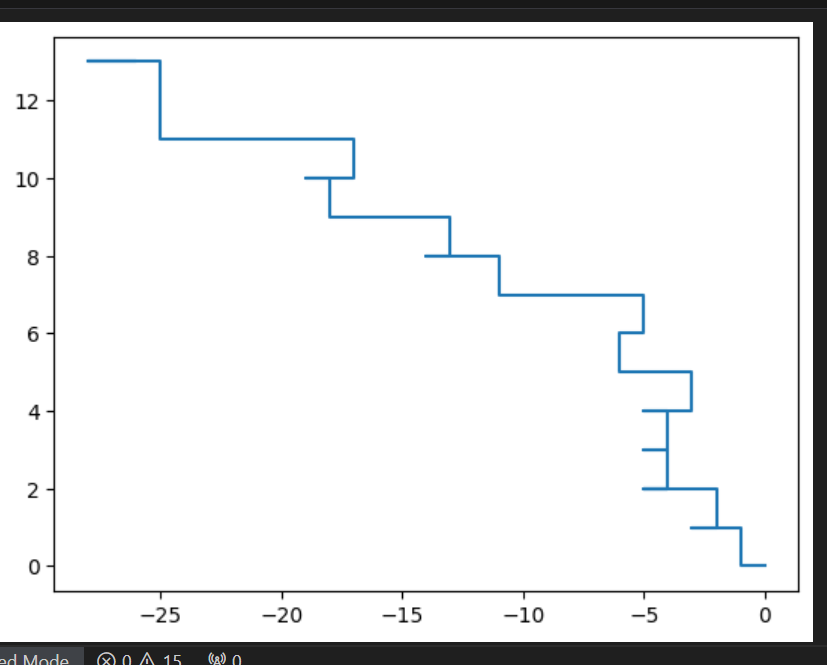
        y.extend([y[i]])

    else:

        x.extend([x[i]-1])

        y.extend([y[i]])

**GRAPH 4;**



import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

prob = [0.95, 0.26 ,0.87]

x=[0]

y=[0]

F = [0, 1, 2, 3, 4]

L = [5, 6, 7]

R = [8, 9]

for i in range(len(RN)):

    if RN[i] in F:

        x.extend([x[i]])

        y.extend([y[i]+1])

    elif RN[i] in R:

        x.extend([x[i]+1])

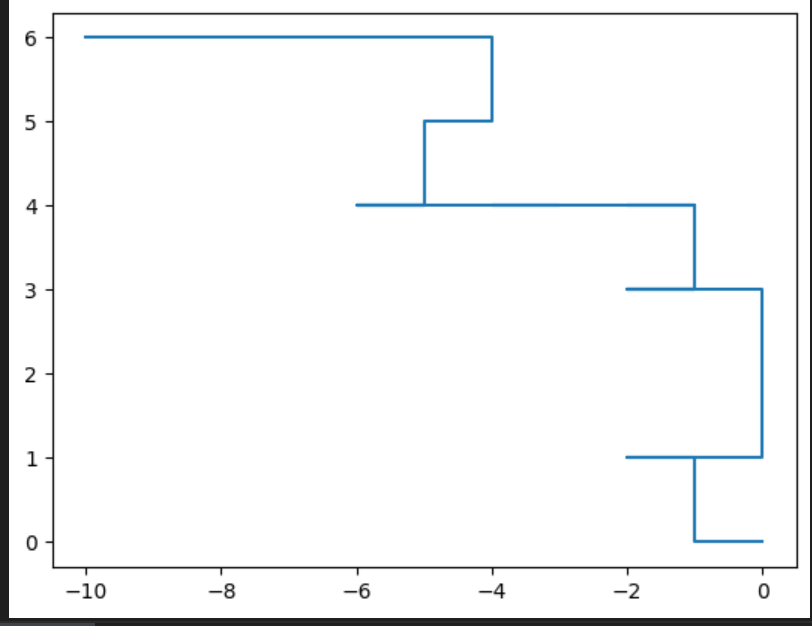
        y.extend([y[i]])

    else:

        x.extend([x[i]-1])

        y.extend([y[i]])

**GRAPH 5:**



import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

prob = [0.87, 0.50 ,0.70, 0.87, 0.87, 0.23, 0.23]

RN = [6, 2,10, 8, 8, 3, 4, 7, 112, 8, 11, 9, 2, 23, 8, 134, 7, 54, 8, 32, 7, 32, 9, 78, 8, 3, 8, 1, 12, 54, 34, 76, 87, 87]

x=[0]

y=[0]

F = [0, 1, 2, 3, 4]

L = [5, 6, 7]

R = [8, 9]

for i in range(len(RN)):

    if RN[i] in F:

        x.extend([x[i]])

        y.extend([y[i]+1])

    elif RN[i] in R:

        x.extend([x[i]+1])

        y.extend([y[i]])

    else:

        x.extend([x[i]-1])

        y.extend([y[i]])

data={"X Cor":x , "Y Cor":y}

df=pd.DataFrame(data=data)