Problem 1 Presentation Outline

**Slide Title: Problem 1 Overview**

* In Part A, we used the random number generator to determine the direction of the walker's motion. We used two for loops one for the number of walks and one for the number of steps in a single random walk.
* In Part B, we used the polyfit function to fit a linear function to the mean square distance data.

**Slide Title: Part A (1/3)**

n = 100

m = 10000.0

x2ave=np.asarray([0.0]\*n)

y2ave=np.copy(x2ave)

xAve=np.copy(x2ave)

dist2=np.copy(x2ave)

**Slide Title: Part A (2/3)**

for j in range(int(m)):

x = 0

y = 0

for i in range(n):

r = random.random()

if r <= 0.25:

x += 1

elif r <= 0.5:

x -= 1

elif r <= 0.75:

y += 1

else:

y -= 1

xAve[i] += x

x2ave[i] += x \*\* 2

dist2[i] += x \*\* 2 + y \*\* 2

**Slide Title: Part A (3/3)**

xAve /= m

x2ave /= m

dist2 /= m

distNew = dist2[3:100]

xAveNew = xAve[3:100]

x2aveNew = x2ave[3:100]

**Slide Title: Part B**

steps=np.arange(4, n + 1, 1)

coefficients=np.polyfit(steps, distNew, 1)

slope=coefficients[0]

diffCoeff=slope/2

eq=np.poly1d(coefficients)

eqSteps=eq(steps)

**Slide Title: <xn> Plot**

* Insert plot of <xn> from Part A

**Slide Title: <xn2> Plot**

* Insert plot of <xn2> from Part A

**Slide Title: r2 Plot**

* Insert plot of r2 from Part B