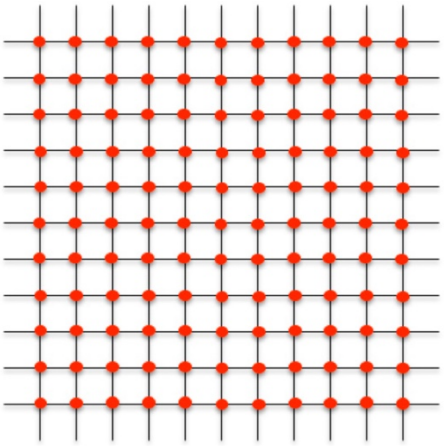
CSE 3504: Probabilistic Analysis of Computer Systems
Spring 2017
Project – Part 1

Delivery deadline: Thursday, March 23rd, 2017.
Goal: To practice calculating basic probability metrics (mean, median, variance, standard deviation)
Prerequisites: some programming knowledge.
Group Size: 2. **Programming Language**: Any

Background: Two-dimensional Random Walks

The integer lattice $Z \times Z$ consists of all points (m, n) in the plane for which m, n are integers.



If we start at point (0, 0), the probability of moving in the first step will be:

$P\big((0,0) \to (1,0)\big) = \frac{1}{4}$ $P\big((0,0) \to (-1,0)\big) = \frac{1}{4}$

$P\big((0,0) \to (0,1)\big) = \frac{1}{4}$ $P\big((0,0) \to (0,-1)\big) = \frac{1}{4}$

A sequence of steps on a lattice $Z \times Z$ is called a simple random walk. Below is an example of it.

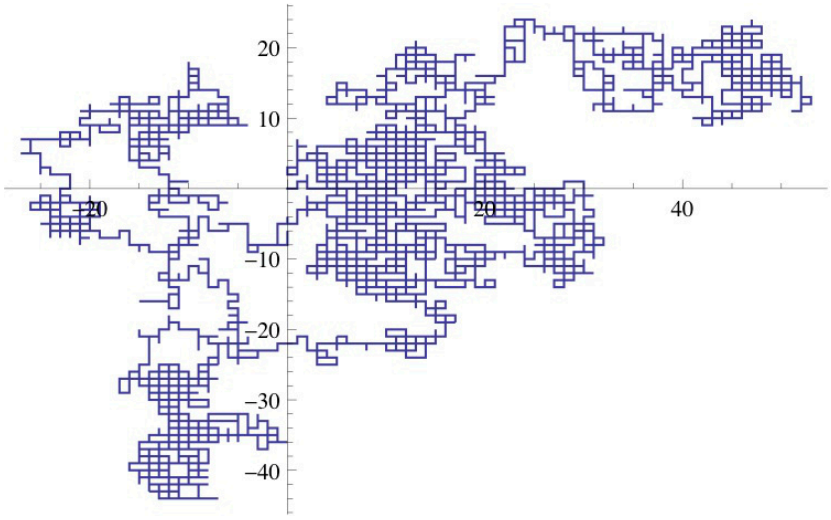


Figure 1 - A simple walk of 5,000 steps. Remember that a step cannot be taken diagonally. Only east, west, north and south directions are allowed.

Other than the simple random walk, there is another type called Self-Avoiding Walk (SAW). In a self-avoiding random walk, you are not allowed to visit the same place more than once. That means taking the first step from point (0,0) has four directions. To take the second step, there are only three directions to choose from. Subsequent steps must first check if the next step is toward a point visited before or not. If it was visited, the walker must choose another step direction (memory property).

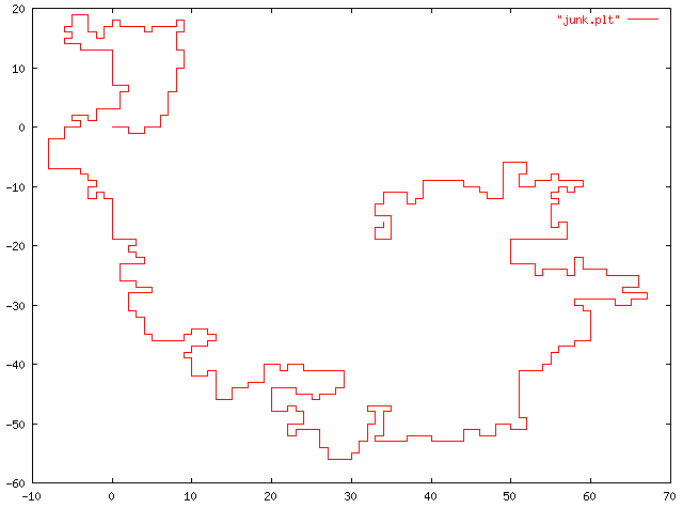


Figure 2 - Example of a Self-Avoiding Walk (SAW)

Submission Instructions:

- On the online submission form on HuskyCT, submit the following:
  1. A one PDF or Word document that has the answers to the questions below.
  2. Two source code files named as following:
     a. Practice1.<any_extension> (e.g. practice1.m or practice1.java)
     b. Practice2.<any_extension> (e.g. practice2.m or practice2.java)
- You may compress the three files in one .zip or .rar file.
- Write the two students names on the PDF or Word document.
- Only one of the two students should submit the materials. There is no need for redundancy.

Practice 1 (30 points):

Use your programming knowledge to simulate a self-avoiding random walk in a finite lattice of 100×100, where the starting point (0, 0) is exactly in the middle of the lattice[1]. The simulation stops when the walker is trapped. A trap occurs when at any point in the path, there is no option available for the next step as all available directions have been visited before or out of the borders of the lattice.

1. Calculate the path length (the number of steps travelled until being trapped).
2. Run the simulation 100 times and calculate the path length at each time. Calculate:
    a. The average path length.
    b. The standard deviation of the path length.
3. Increase the lattice size to 1000×1000 and answer the same questions as in point 2.

Practice 2 (40 points):

Simulate a self-avoiding random walk of 5,000 steps in an infinite lattice (no border constraints from any direction)[2].

1. What is the distance between the start point and the end point?
   (distance = $|x_{start} - x_{end}| + |y_{start} - y_{end}|$)
2. What is the lattice size?
   (size = $(x_{max} - x_{min})\times(y_{max} - y_{\min})$)
3. Repeat the simulation 10 times and calculate:
    a. Distance average (the mean), median, variance and standard deviation.
    b. Lattice size average (the mean), median, variance and standard deviation.
4. Run the simulation 100 times and answer the same questions as in point 3.
5. Run the simulation 1000 times and answer the same questions as in point 3.
6. Write a comment on how the results change with changing the number of simulations.

Hint: write code that doesn't require you to manually run the simulation every time.

Question 3 (30 points):

Self-avoiding random walks have many applications in different areas. Research one application and write a one-page description on it. Discuss how this particular application varies from the simulations you just did in the previous two practices.

[1] i.e. from the start point, the walker can go up to 50 steps only in any of the four directions.

[2] If the walker is trapped, the simulation stops (in that case the walk will have a length less than 5,000 steps).

Helper Pseudocode

The below pseudocode is just a guidance for the simulation logic. You should inject your calculations appropriately. You should also adapt the code to the programming language you will use.

Practice 1:

1. Start at (0, 0). Mark it visited.
2. Enumerate all not-visited directions in a random order (Check for borders).
3. Next step = pick next direction.
4. If next step is null, stop simulation.
5. If next step is visited, go to step 3.
6. Current point = next step. Mark current point as visited.
7. Go to step 2.

Practice 2:

1. Number of steps = 0
2. Start at (0, 0). Mark it visited.
3. If number of steps == 500, stop simulation.
4. Enumerate all not-visited directions in a random order (Check for borders).
5. Next step = pick next direction.
6. If next step is null, stop simulation.
7. If next step is visited, go to step 5.
8. Current point = next step. Mark current point as visited. Increment number of steps.
9. Go to step 3.