Queens College, CUNY, Department of Computer Science

Software Engineering

CSCI 370

Fall 2018

Instructor: Dr. Sateesh Mane

due Sunday December 16, 2018

5 Project 5c

- This document describes a mathematical calculation involving a lot of computation.
- To reduce the overall computation time, the application should be multi-threaded, so that the computations can be performed in parallel.
- You are responsible for configuring how your application manages the threads.
- You are responsible to design your program code to perform the computations in parallel.
- This project does not require a GUI or a database.
- The application will be tested by running it on the Mars server.

5.1 Random walks

- Let x be a variable which takes integer values.
- The variable x executes a random walk as follows.
  1. Define positive integers u and d, where $d > u$, e.g. $u = 1$ and $d = 2$.
  2. At each time step, the value of x goes up by u or down by d.
  3. The probability is $\frac{1}{2}$ for a step in either direction.
  4. The mathematical formula is as follows:

$$x = \begin{cases} x + u & (\text{prob} = \frac{1}{2})\,, \\ \\ x - d & (\text{prob} = \frac{1}{2})\,. \end{cases} \tag{5.1.1}$$

  5. This is an asymmetric random walk: the up/down steps have unequal size.
  6. This random walk has a net negative or downward drift because $d > u$.
  7. *The more usual model is to have equal steps ± 1 and unequal probabilities for the up and down steps. We are doing something different.*
- We run a random walk simulation as follows.
  1. Measure the "time" in integer steps $n = 0, 1, 2, \ldots$
  2. Initialize $x = k$, where $k > 0$ is a positive integer, so $x = k$ at $n = 0$.
  3. Then at $n = 1$ the value of x is either $k + u$ else $k - d$, with equal probability.
  4. Run a loop over n and increment the value of x at each time step.
  5. Because of the downward drift, the value of x will eventually become zero or negative.
  6. **Terminate the random walk as soon as $x \leq 0$.**
  7. **The value of n at which this happens is called the first stopping time.**
  8. It is also known as the *first hitting time* or *first passage time*.

5.2 Probability distribution of first stopping time

- We construct the probability distribution of the first stopping time as follows.

  1. Run a total of M random walk simulations.
  2. For each random walk, record the value of n as soon as $x \leq 0$.
  3. Construct a histogram of the values the first stopping time.
  4. Normalize the histogram so that the total area equals 1.
  5. Let the heights in the bins be h_n, $n = 0, 1, 2, \ldots$.
  6. Then we want the sum of all the heights to equal 1:

$$\sum_n h_n = 1\,. \tag{5.2.2}$$

  7. Then the histogram will display the probability distribution of the first stopping time.
  8. Clearly, if M is large, the results will be more accurate (more samples).

- Begin with $M = 10^4$ or 10^6, for example, for testing.
- **For the project, we want a sample size of $M \geq 10^9$ (one billion) random walks.**
- This is a large sample, hence the computations should be run in parallel in multiple threads.
- It is your responsibility to write a simulation algorithm for each random walk.
- It is your responsibility to manage the threads and compute the histogram.

5.3 Histogram

- **The histogram should be written to file.**
  1. The data in the file should consist of two columns n and h_n.
  2. Let $n_{\max}$ be the largest value of n of the program output.
  3. Then the output file should contain $n_{\max}$ rows, from $n = 1$ to $n = n_{\max}$.
  4. **If a bin is empty, then print $h_n = 0$ for that bin.**
  5. Obviously the bins will be empty for $1 \le n < k/2$.
  6. The output file will be uploaded to Excel (for example).
  7. The histogram will be charted using Excel, or some other graphing tool.
- An example output (a graph rather than a histogram) is displayed in Fig. 1, for $k = 100$, $u = 1$, $d = 2$ and a sample size of $M = 10^7$.
- Despite appearances, it it is actually one probability distribution, it contains two subsets.

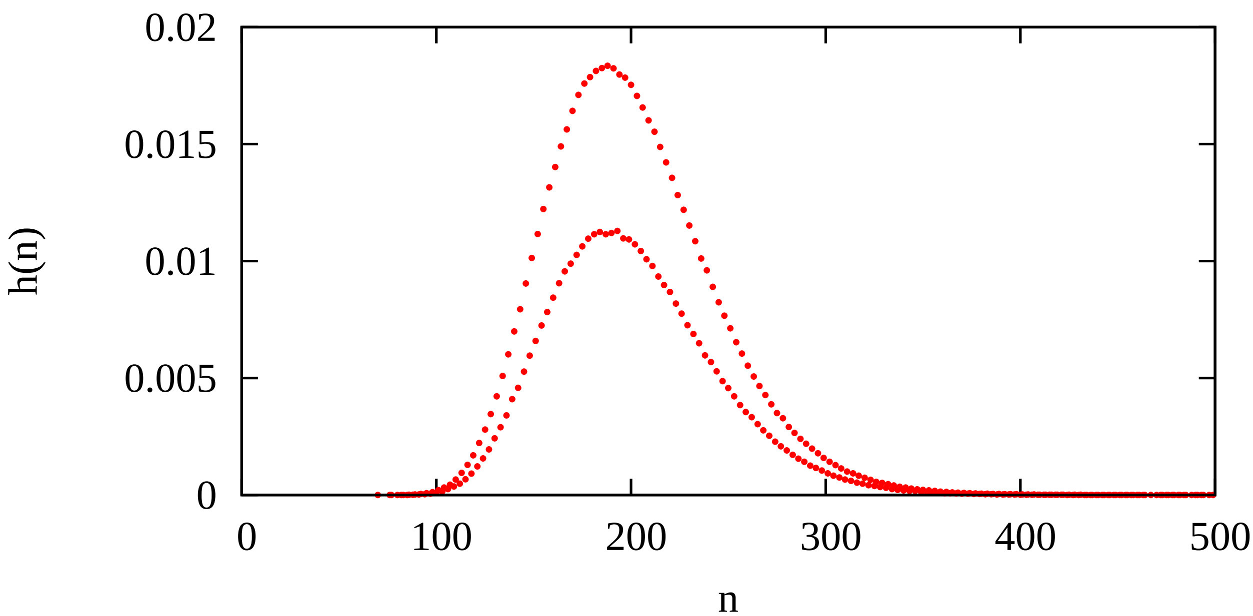

Figure 1: Graph of probability distribution of first stopping times for $k = 100$, $u = 1$, $d = 2$ and $M = 10^7$.

5.4 Mean and variance

- *This should be easy.*
- Write a (different) program to read the histogram file.
- The program should compute the mean and variance as follows.
- The mean μ is given by the following formula:

$$\mu = \sum_{n=1}^{n_{\max}} n\, h_n \,. \tag{5.4.3}$$

- The variance σ^2 is given by the following formula:

$$\sigma^2 = \biggl(\sum_{n=1}^{n_{\max}} n^2\, h_n \biggr) - \mu^2 \,. \tag{5.4.4}$$

- If you do your work correctly, you should find that for large k (and fixed values of u and d)

$$\mu = O(k)\,, \qquad \sigma^2 = O(k)\,. \tag{5.4.5}$$

- In other words, the standard deviation σ is of order $O(\sqrt{k})$.
- Graphs of μ and σ^2 are plotted in Figs. 2 and 3, respectively.
  Straight line fits to the data are also plotted.
- To obtain the above results you will have to run multiple simulations and obtain histograms for several values of k.
- **It is therefore essential to optimize the running time of your simulation program.**

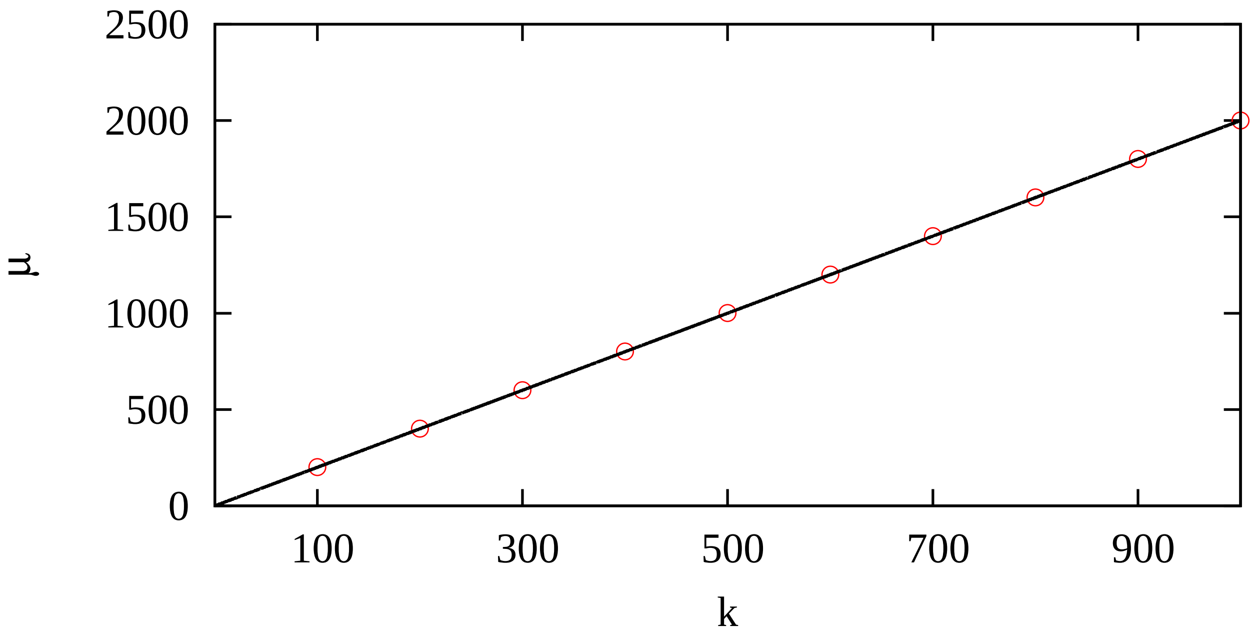

Figure 2: Graph of the mean μ of the first stopping time vs. k, for $u = 1$ and $d = 2$. The straight line is $\mu = 2k$.

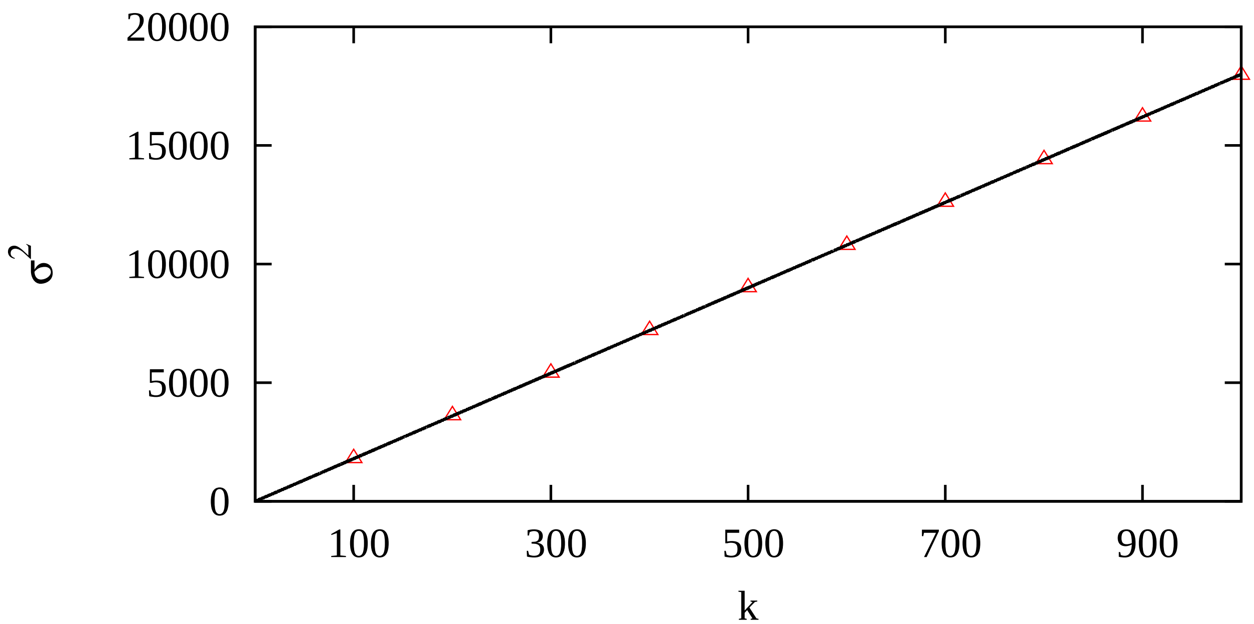

Figure 3: Graph of the variance σ^2 of the first stopping time vs. k, for $u = 1$ and $d = 2$. The straight line is $\sigma^2 = 18k$.

5.5 Project report

- Your project zip archive must contain all your program source code.
  1. Program for random walk simulations and multi-threading.
  2. Program to calculate the mean and variance.
- Your project report must contain a description of your program architecture.
  It is your responsibility to explain the architecture clearly.
- Your project report must contain screenshots of relevant output.
  It is your responsibility how to offer a clear and comprehensive set of graphs.