Tasks labeled by the 🏵 sign are optional and are worth extra credits.

General information

Contact: zsolt.lazar@ubbcluj.ro
Course website: MS Teams
Assignment: Submitted assignment should be archived folder with its name containing the assignment number and title of the assignment, e.g., `02_Random_walk_part_1`. The folder should be uploaded as a compressed archive containing the student's name, e.g., `Smith_John_02_Random_walk_part_1.zip`.

Requirements

Reports should include:

1. Students name
2. Title
3. Statement of the problem
4. Hypotheses
5. Strategy/method
6. Conclusions
7. Complete compilation/installation/execution instructions. Specifying dependencies are appreciated.
8. Results from demo runs including screen captures of charts, etc. - where it applies - should be included.
9. Detail the parameters used for the demonstrated runs (on the Figures)
10. Describe the environment(s) where the testing was successful (compilers, libraries, environments,...)
11. Have a list of source files attached.

Source code should be readable:

12. properly segmented, consistently indented
13. generously commented (including the description of the approach)
14. self-explanatory naming of variables

All programing tasks have to be implemented in C/C++. Some of the tasks are to be implemented ALSO in a language with vector/matrix operation capabilities like Python or Matlab. These are marked with a P/M sign.
In these cases emphasis is on employing builtin functions of the language and avoiding loops. The algorithms do not have to be fully equivalent to their C variant but rather focusing on simplicity as primary goal.
Difference between the two approaches (C vs. Python/Matlab), if any, together with benefits, drawbacks and limitations should be discussed.
Submission beyond deadline is possible but penalized.

1 Warm up

Deadline: 11.03

1. Dice throwing: show that the distribution is uniform (1/6), the variance goes to zero. How fast? (P/M)
2. Demonstrate the central limit theorem using the total points obtained when throwing simultaneously with several dice (P/M).

   Hint: Calculate the sum of N throws, repeat the experiment and look at the distribution of the sums. In Python one can use the `hist` or `histogram` functions. Try to show that for large N it is a normal distribution.

3. The secretary problem: we have to choose the best possible candidate from N applicants so that we can only interview once (or not at all) each applicant and have to decide on the spot. The strategy is that we refuse the first k applicants irrespective of their performance and then we accept the first applicant that outperforms the best of the first k. The rest are sent home without an interview. If none do we have to employ the very last applicant.

   (a) What is the k/N ratio that maximizes the chance that we employ the best applicant? (P/M)

   (b) (+10%) Provide an analytical proof of the above result

   (c) What is the k/N ratio that maximizes the average quality (rank) of the accepted applicant. (P/M)

4. Plain random walk (no self-avoidance) in one and two dimensions. Study the following:

   (a) the probability distribution of the final position in one dimension (P/M). Show that it is a normal distribution.

   (b) $\langle d^2(N) \rangle$ dependence for the plain random walk in two dimensions. Use a square lattice.

   (c) the effect of the direction set (all directions have the same probability). Study them both analytically and test numerically:

      i. four main directions (up, down, left, right)(P/M (+10%)), and halt (P/M).

      ii. four diagonals (up-right, up-left, down-right, down-left) (P/M)

      iii. all eight directions (P/M (+10%))

      iv. all eight directions + halt (no movement) (P/M)