

Programowanie II R

Zadania – seria 8.

Generatory liczb pseudolosowych

W poprzednich zadaniach do generowania liczb pseudolosowych używaliśmy funkcji `rand()` z biblioteki `<cstdlib>`. We współczesnym C++ zaleca się generowanie liczb losowych za pomocą narzędzi z pliku nagłówkowego `<random>`. Zdefiniowane są w nim generatory liczb całkowitych o jednorodnym rozkładzie (silniki) oraz obiekty pozwalające zamienić liczby wygenerowane przez silnik na liczby o zadanym rozkładzie prawdopodobieństwa (rozkłady).

Zadanie: Centralne twierdzenie graniczne

Wynik pomiaru X jest często efektem sumowania się bardzo wielu niezależnych czynników losowych x_1, x_2, \dots, x_n . Zgodnie z centralnym twierdzeniem granicznym, niezależnie od rozkładu prawdopodobieństwa poszczególnych składowych x_i , rozkład ich średniej arytmetycznej dąży do rozkładu normalnego (Gaussa) wraz ze wzrostem n . Dodatkowo, jeśli odchylenie standardowe pojedynczej zmiennej wynosi σ_x , to odchylenie standardowe średniej σ_X powinno wynosić:

$$\sigma_X = \frac{\sigma_x}{\sqrt{n}} \quad (1)$$

Napisz program, który przeprowadzi symulację 10^5 pomiarów. Wynik każdego pomiaru (X) to średnia z n liczb losowych x_i .

1. Wykorzystaj silnik Mersenne Twister `std::mt19937`.
2. Do wytworzenia liczby losowej służącej do inicjalizacji (seedowania) silnika użyj klasy `std::random_device`.
3. Przeprowadź symulację dla trzech rozkładów x_i :
 - Jednorodny (`std::uniform_real_distribution`).
 - Wykładniczy (`std::exponential_distribution`).
 - Dwumianowy (`std::bernoulli_distribution`).
4. Przeprowadź symulację dla trzech wartości n : 10, 100, 1000.
5. Oblicz średnią oraz odchylenie standardowe zmiennych X i x_i . Sprawdź, czy odchylenie standardowe pomiaru zgadza się z przewidywanym przez centralne twierdzenie graniczne.
6. Narysuj histogram zmiennych X i x_i . Można do tego wykorzystać funkcję z pliku `draw_histogram.hpp`.

Opracowanie: Piotr Dziekan