

# Generatory liczb pseudolosowych

Paweł Prochot

## 1. Opis projektu

Celem niniejszego projektu jest zaimplementowanie generatora liczb losowych **G** o rozkładzie *równomiernym*, oraz na jego bazie, generatora **U** z rozkładem *jednostajnym* na przedziale **(a, b)**.

Korzystając z generatora **U**, tworzone są generatory dla najpopularniejszych rozkładów tj.

- *Bernoulliego (dwupunktowego)*
- *Dwumianowego*
- *Poissona*
- *Wykładniczego*
- *Normalnego*

## 2. Informacje dotyczące implementacji

Generatory zostały napisane w języku C++.

Deklaracje generatorów znajdują się w folderze *Header Files*, a ich implementacje w *Source Files*.

## 3. (G) Generator liczb całkowitych o rozkładzie równomiernym

Generator **G** to liniowy generator multiplikatywny.

Generowana liczba jest postaci:

$$X_{n+1} = (aX_n + c) \bmod m$$

Podstawową implementację generatora można otrzymać za pomocą metody *defaultGenerator()*, jej parametry to:

$$m = 2^{31} - 1$$

$$a = 7^5$$

$$c = 0$$

Seed do generatora jest tworzony za pomocą metody *initalizeSeed()*, która przy użyciu aktualnej daty w systemie tworzy liczbę według wzoru:

$$s + 60 \left( \min + 60 \left( g + 24(d - 1 + 31(m - 1 + 12r)) \right) \right)$$

$s$  – sekundy,  $\min$  – minuty,  $g$  – godziny,  $d$  – dzień,  $m$  – miesiąc,  $r$  – rok

Przy czym ostatni bit tej liczby jest zawsze zamieniany na 1, tak aby liczba była nieparzysta.

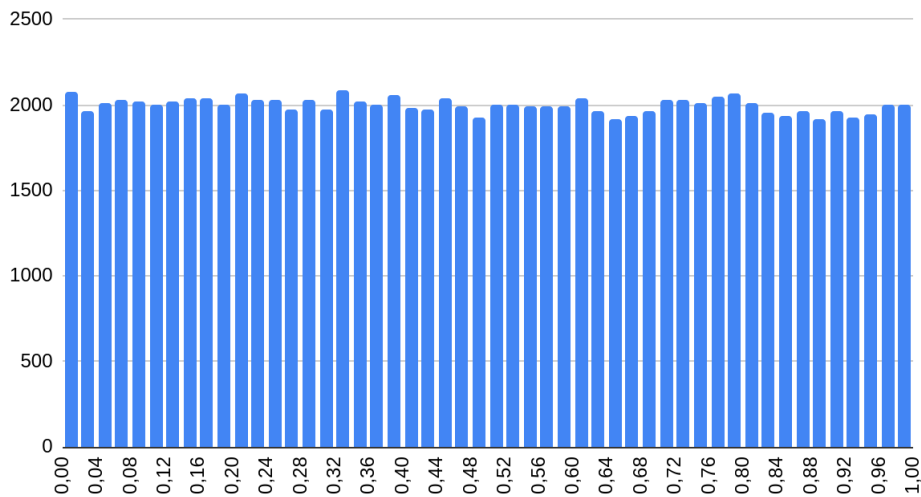
#### 4. (U) Generator liczb z rozkładu jednostajnego

Generator **U** korzysta w swojej implementacji z generatora **G**.

Liczby przez niego generowane należą do przedziału od **(a, b)**, jednakże w implementacjach kolejnych generatorów korzystana jest jego wersja na przedziale **(0, 1)**.

Wygenerowana liczba jest postaci  $x * \frac{b-a}{m} + a$ , gdzie  $m$  to parametr z generatora **G**.

Rozkład dla generatora U(0, 1) dla 100 000 liczb

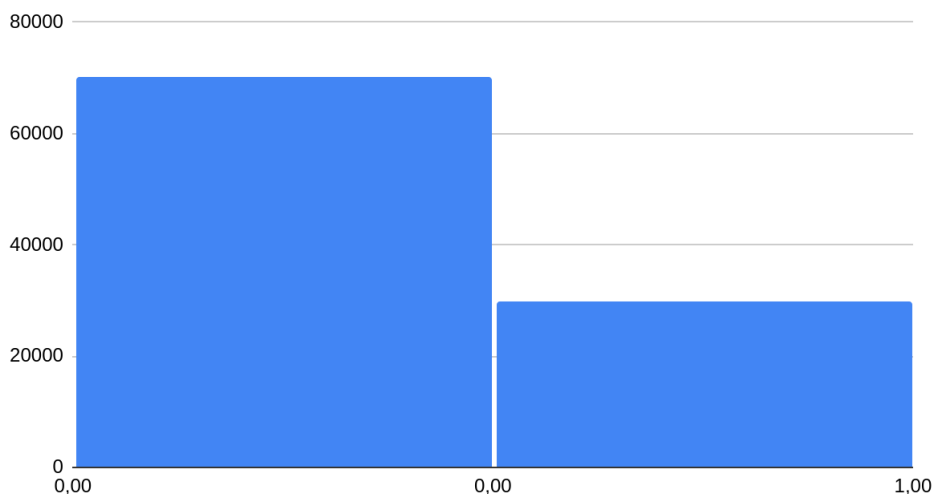


#### 5. (B) Generator o rozkładzie Bernoulliego

Generator przyjmuje pewne prawdopodobieństwo **p**, oraz generator **U(0, 1)**.

Na początku generowana jest liczba z **U**, którą następnie porównuje się z prawdopodobieństwem **p** i jeśli liczba jest mniejsza bądź równa to zwracane jest 1, w p.p. 0.

Rozkład dla generatora B z  $p = 0.3$  dla 100 000 prób

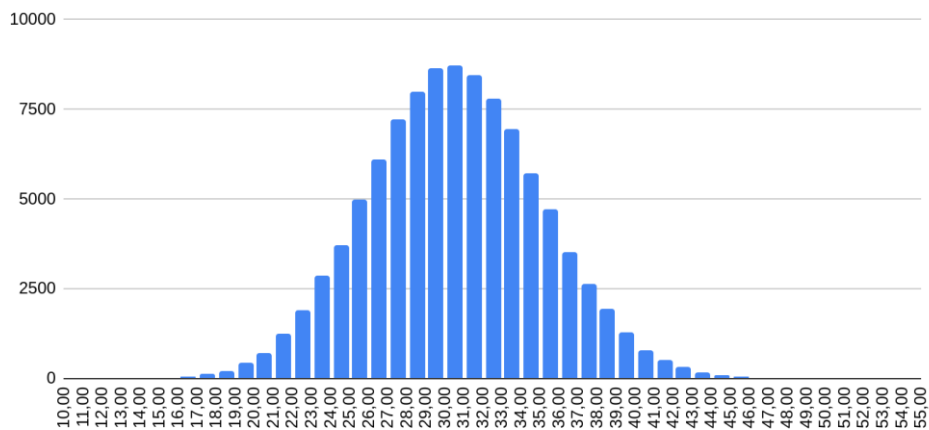


#### 6. (D) Generator o rozkładzie dwumianowym

Generator przyjmuje pewne prawdopodobieństwo **p**, oraz liczbę prób **n**.

Algorytm generuje **n** liczb z rozkładu **U(0, 1)** i sprawdza ile liczb jest mniejszych bądź równych od **p**.

Rozkład dla generatora D z parametrami  $n=100$  i  $p=0.3$  dla 100 000 prób



## 7. (P) Generator o rozkładzie Poissona

Korzystając z lematu 3.9 z *Komputerowe generatory liczb losowych*, tj.

Jeżeli  $a_1, \dots, a_n$  to liczby o jednakowym rozkładzie wykładniczym  $U(0, 1)$ , to zmienna losowa  $U = \min\{j: \sum_{i=0}^j a_i > \lambda\}$  ma rozkład Poissona z parametrem  $\lambda$ .

Dzięki temu lematowi, jesteśmy w stanie skonstruować algorytm generujący rozkład Poissona.

### ALGORYTM 3.27

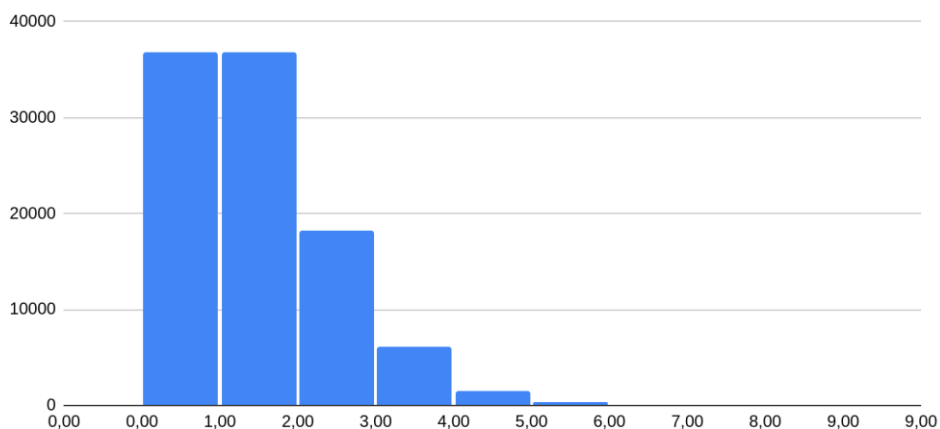
$q = e^{-\lambda}$ ,  $X = 0$ ,  $S = q$ ,  $P = q$

Generuj  $U$  o rozkładzie  $U(0, 1)$

While  $U > S$  do  $X = X + 1$ ,  $P = P * \lambda / X$ ,  $S = S + P$

Return  $X$

Rozkład dla generatora P z parametrami  $\lambda=1$  dla 100 000 prób

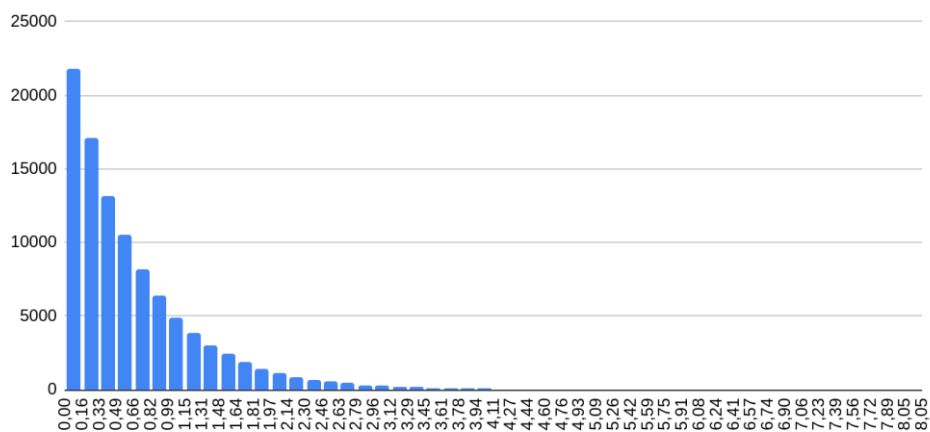


## 8. (W) Generator o rozkładzie wykładniczym

Do wygenerowania liczb o rozkładzie wykładniczym z parametrem  $\lambda$ , korzystamy z generatora

$U(0, 1)$  i obliczona wartość jest postaci:  $T = -\frac{\ln(u)}{\lambda}$ .

Rozkład dla generatora W z parametrem lambda=1,5 dla 100 000 prób

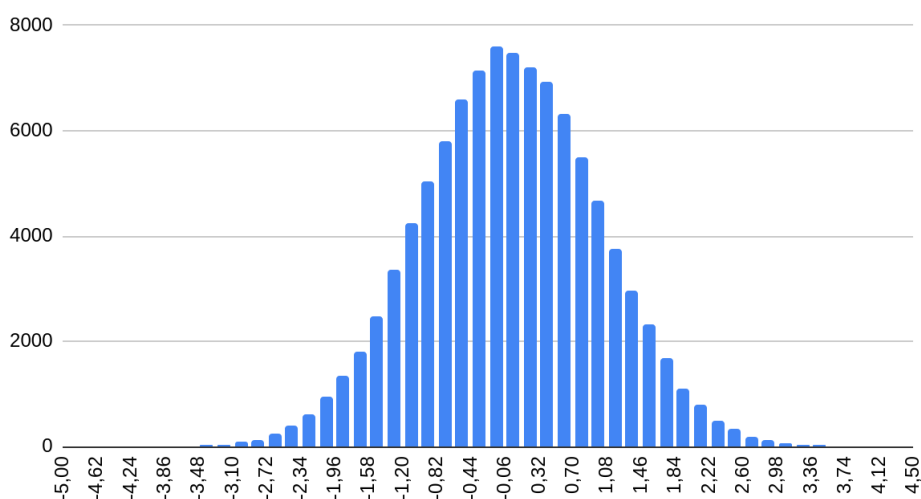


### 9. (N) Generator o rozkładzie normalnym

Aby wygenerować liczbę z rozkładu normalnego, korzystamy z *transformacji Boxa-Mullera*.

Losujemy dwie liczby  $u$  i  $v$  z rozkładu  $U(0, 1)$ , a następnie liczba postaci  $Z = \sqrt{-2 \ln u} \cos(2\pi v)$ , jest liczbą z rozkładu normalnego.

Rozkład dla generatora N dla 100 000 prób



### 10. Źródła

[Exponential distribution - Wikipedia](#)

[Normal distribution - Wikipedia](#)

[Box-Muller transform - Wikipedia](#)