Sieci Mobilne i Bezprzewodowe
laboratorium 2
Modelowanie zdarzeń dyskretnych

Plan laboratorium

- Generatory liczb pseudolosowych dla rozkładów dyskretnych:
  - Generator liczb o rozkładzie równomiernym
  - Generator liczb o rozkładzie dwumianowym (Bernoulliego)
  - Generator liczb o rozkładzie Poissona,
  - Generator liczb o rozkładzie geometrycznym
  - Generator liczb o równym rozbiciu przedziału (równomierny)

Generatory liczb pseudolosowych dla rozkładów dyskretnych

Liczby losowe i generatory liczb losowych

- Liczba losowa – konkretna wartość przyjmowana przez zmienną losową (**nieprzewidywalna!**).
- Sekwencja liczb prawdziwie losowych jest nieprzewidywalna, a zatem niereprodukowalna!
- Generatory liczb losowych (RNG) – generatory fizyczne:
  - „mechaniczne” – np. rzut monetą, losowanie z urny, ruletka itp.
  - oparte o procesy fizyczne – np. szum w urządzeniach elektronicznych (szczególnie tzw. szum biały), rozpad radioaktywny, promienie kosmiczne itp.
- Wady generatorów fizycznych:
  - zbyt wolne dla typowych potrzeb obliczeniowych (szczególnie „mechaniczne”);
  - problemy ze stabilności – szczególnie generatory oparte o procesy fizyczne, np. niewielka zmiana warunków fizycznych źródła lub otoczenia może spowodować istotne zmiany własności probabilistycznych otrzymanych liczb co wymaga dodatkowych urządzeń testujących i korygujących.
- Dawniej w użytku były tablice liczb losowych – niezbyt praktyczne!

Liczby pseudolosowe i generatory liczb pseudolosowych

- Liczby pseudolosowe – liczby generowane według ścisłej formuły matematycznej (zatem reprodukowalne, a więc w ogóle nielosowe w sensie matematycznym), ale mające „wygląd” losowości, tzn. ich własności statystyczne są bardzo bliskie własnościom liczb prawdziwie losowych (ktoś kto nie zna formuły, według której są generowane, nie powinien być w stanie stwierdzić´, że nie są to liczby prawdziwie losowe).
- Generatory liczb pseudolosowych (PRNG) – generatory matematyczne (programowe):
  - dobre własności statystyczne generowanych liczb,
  - łatwość użycia (proste, szybkie, wygodne, . . .).
- Wyparły prawie zupełnie generatory fizyczne!
- Dlatego też liczby pseudolosowe są nazywane po prostu liczbami losowymi (ang. *Random numbers), a matematyczne algorytmy do ich otrzymywania nazywane są generatorami liczb* losowych (ang. *random number generators – RNG)*.

Generator liczb o rozkładzie równomiernym U(0,1)

Algorytm 1:

Generuje liczby losowe o rozkładzie równomiernym z przedziału (0,1) przy pomocy równania:

$$x_{n+1} = (ax_n + b) \bmod c$$

gdzie 0 < a, b < c oraz są szczególnie dobranymi liczbami, np.: $a=7^5=16807$, b=0 oraz $c=2^{31}-1=2147483647$, oraz x_0 jest ziarnem generatora.

Uwaga: Liczby otrzymane w ten sposób $\in\{0, 1, \ldots, 2^{31}-2\}$.

Liczbę losową `X ∈ (0,1)` otrzymujemy: **$X = x_{n+1}/c$**

Generator liczb o rozkładzie równomiernym $U(0,1)$

Algorytm 2:

Generuje liczby losowe o rozkładzie równomiernym z przedziału (0, 1) przy pomocy schematu:

1. Losuj R – liczbę całkowitą z zakresu (0, Range)
2. $X = R / Range$

Algorytm 3:

Generuje liczby losowe o rozkładzie równomiernym z przedziału (a, b) przy pomocy schematu:

1. Losuj R – liczbę całkowitą z zakresu (0, Range)
2. $X = a + (b - a) R / Range$

Generator liczb o rozkładzie dwumianowym (Bernoulliego)

$$P(X=k)=P(X(k;N)=P_k=\binom{N}{k}p^k q^{N-k}$$

Algorytm 1:

```
X = 0;
for (i=0; i<N; i++){
  U = GenU(0,1);
  if (U <= p) X++;
}
return X;
```

Wymaga odwołania się do generatora U(0,1) liczb losowych o rozkładzie równomiernym,
tzn. GenU – generowanie zmiennej losowej U z rozkładu równomiernego U(0,1)

- Wada: Wymaga wielu liczb losowych.

Generator liczb o rozkładzie dwumianowym (Bernoulliego) c.d.

Jeżeli N nie jest bardzo duże, to może opłacać się stablicowanie dystrybuanty, tzn. obliczenie:

$$p_k = \sum_{i=0}^{k} \mathcal{P}\{X = i\}$$

i skorzystanie z następującego algorytmu, który wymaga tylko jednej liczby losowej:

Algorytm 2:

```
X = 0;
U = GenU(0,1);
while (U<=p_X) X++;
return X;
```

Także wymaga odwołania się do generatora U(0,1) liczb losowych o rozkładzie równomiernym,
tzn. GenU – generowanie zmiennej losowej U z rozkładu równomiernego U(0,1)

Generator liczb o rozkładzie dwumianowym (Bernoulliego) c.d.

Algorytm 3: (potrzebna tylko jedna liczba losowa!)

```
X = 0;
U = GenU(0,1);
  for (i=0; i<N; i++){
      if (U<=p) {X++; U=U/p;}
      else U=(1-U)/(1-p);
  }
return X;
```

Także wymaga odwołania się do generatora U(0,1) liczb losowych o rozkładzie równomiernym, tzn. GenU – generowanie zmiennej losowej U z rozkładu równomiernego U(0,1)

Dowód i szczegóły, patrz:

R. Wieczorkowski, R. Zieliński, *Komputerowe generatory liczb losowych*, WNT 1997

Generator liczb o rozkładzie Poissona

$$P(X=k)=e^{-\lambda}\frac{\lambda^k}{k!},\quad \lambda>0,\quad k=0,1,2,\ldots$$

Algorytm 1:

```
X = -1; S = 0;
while (S<=lambda){
  Y = GenE(0,1);
  S=S+Y; X++;
}
return X;
```

Wymaga odwołania się do generatora E(0,1) liczb losowych o rozkładzie wykładniczym, tzn. GenE – generowanie zmiennej losowej Y z rozkładu wykładniczego E(0,1)

Wada: Wymaga wielu liczb losowych.

Generator liczb o rozkładzie Poissona c.d.

Algorytm z wykorzystaniem rozkładu równomiernego:

```
Algorytm 2:
X = -1; S = 1;
q = exp(-lambda);
while (S > q) {
  U = GenU(0,1);
  S=S*U; X++;
}
return X;
```

Wymaga odwołania się do generatora U(0,1) liczb losowych o rozkładzie równomiernym, tzn. GenU – generowanie zmiennej losowej U z rozkładu równomiernego U(0,1)

Wada: Wymaga wielu liczb losowych.

Generator liczb o rozkładzie Poissona c.d.

Algorytm 3:

```
X = 0;
q = exp(-lambda);
S = q; P = q;
U = GenU(0,1);
while (U > S) {
  X++;
  P=P*lambda/X;
  S=S+P;
}
return X;
```

Także wymaga odwołania się do generatora U(0,1) liczb losowych o rozkładzie równomiernym, tzn. GenU – generowanie zmiennej losowej U z rozkładu równomiernego U(0,1)

- Wada: Dla dużych wartości λ kumulacja błędów zaokrągleń przy obliczaniu prawdopodobieństw P może prowadzić do niedokładności numerycznych.

Generator liczb o rozkładzie geometrycznym

$$P(X = k) = (1 - p)^{k-1} p, \quad k = 1, 2, 3, \ldots$$

```
Algorytm:
U = GenU(0,1);
X = ln(U) / ln(p);
return X;
```

Wymaga odwołania się do generatora U(0,1) liczb losowych o rozkładzie równomiernym,
tzn. GenU – generowanie zmiennej losowej U z rozkładu równomiernego U(0,1)

Generator liczb o rozkładzie równomiernego rozbicia przedziału

- Przedział (0. 1) dzielimy na K+1 podprzedziałów (binów) $\left(\frac{i-1}{K+1}, \frac{i}{K+1}\right)$ o jednakowej długości i numerujemy je kolejnymi liczbami 1, 2, . . . , K+1.
- Zmienna U∈U(0, 1) wpada do binu o numerze [(K + 1)U+1].
- Tworzymy ciąg: $q_j = \sum_{k=0}^{j} p_k, \quad j = 0, 1, \ldots, K.$
- Tworzymy pomocniczy ciąg liczb:

$$g_i = \max\left\{j : q_j < \frac{i}{K+1}\right\}, \quad i = 1, 2, \ldots, K+1.$$

Algorytm:

```
U = GenU(0,1);
X = g[(int) (K+1)U+1] + 1;
while (q[X-1]>U) X--;
return X;
```

Zadania implementacyjne:

- Utworzyć aplikację generującą rozkłady dyskretne na podstawie generatora liczb U(0, 1), w której:
  - Zaimplementować generatory rozkładu dwumianowego według poznanych algorytmów. Otrzymane rozkłady przedstawić na histogramach,
  - Zaimplementować generatory rozkładu Poissona według poznanych algorytmów i wygenerować rozkłady dla kilku wartości λ. Otrzymane rozkłady przedstawić na histogramach,
  - Zaimplementować generator rozkładu geometrycznego i wygenerować rozkłady dla kilku wartości p. Otrzymane rozkłady przedstawić na histogramach,
  - Zaimplementować generator rozkładu równomiernego rozbicia przedziału. Otrzymane rozkłady przedstawić na histogramach.
- Stworzyć możliwość generowania z ziarnem i bez użycia ziarna.

Przygotować na kolejne zajęcia:

- Rozkład normalny (Gaussa) zmiennej losowej, generowanie rozkładu.
- Zgromadzić informacje na temat modelu generowania ruchu:
  - autostrady,
  - wybuchu,
  - statycznego (stacjonarnego).