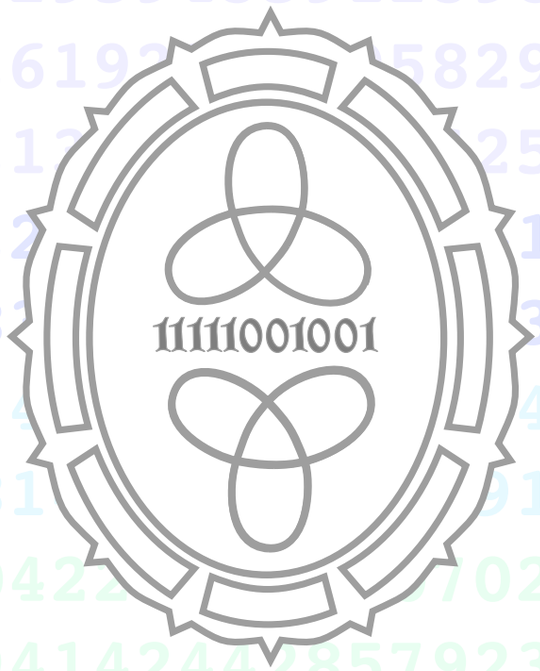
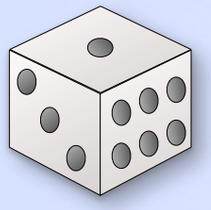


Losowość w informatyce

Czyli czy komputer potrafi rzucać kostką?

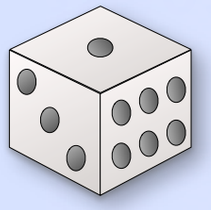
Filip Piękniewski 2006



Co to znaczy “losowy” ?

- Losowość nie jest pojęciem oczywistym!
- Intuicyjne rozumienie losowości jako braku regularności może być mylne!

- Żeby pytać się czy komputer umie generować liczby losowe musimy najpierw wiedzieć o co nam chodzi!
- “*Każdy kto rozważa arytmetyczne metody wytwarzania cyfr losowych jest oczywiście w stanie grzechu*” - John von Neumann (1951)...

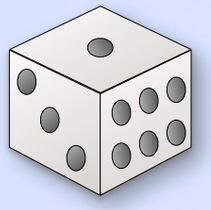


Co to znaczy “losowy” ?

- Czy liczba może być losowa? **NIE**
- Czy liczba 17 jest losowa? **NIE**
- Czy liczba 73254951 jest losowa? **NIE**
- Czy liczby 1,2,3,4,5,6 są losowe? **NIE**

Ściśle losowy może być tylko nieskończony ciąg liczb.

Losowość to nie cecha konkretnej liczby, tylko tego kiedy i w jakim otoczeniu sie pojawia.

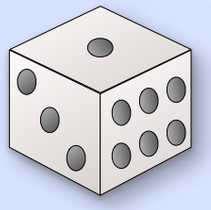


Co to znaczy “losowy”?

- Wyobraźmy sobie, że rzucamy kostką sześcienną. Czego się spodziewamy po ciągu wyników takich losowań?
- Z pewnością każdy z wyników powinien być tak samo prawdopodobny, spodziewamy się zatem, że trójek będzie “mniej więcej” tyle samo co jedynek itp.
- Ale w ciągu:

1 2 3 4 5 6 | 1 2 3 4 5 6 | 1 2 3 4 5 6 | 1 2 3 4 5 6 ...

elementy występują równie często, a jednak zupełnie nie jest on losowy!

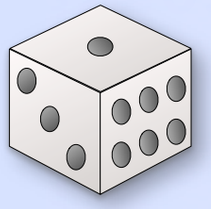


Co to znaczy “losowy”?

- Trzeba zatem wymagać nieco więcej. Intuicja podpowiada, że w losowym ciągu każda z 36 możliwych par elementów powinna występować równie często.
- Tyle, że ciąg:

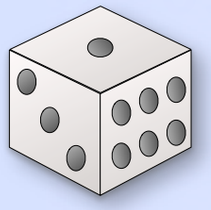
1 1 1 2 1 3 1 4 1 5 1 6 2 1 2 2 2 3 2 4 2 5 2 6 3 1 3 2
3 3 3 4 3 5 3 6 4 1 4 2 4 3 4 4 4 5 4 6 5 1 5 2 5 3 5 4
5 5 5 6 6 1 6 2 6 3 6 4 6 5 6 6 1 1 1 2 1 3 ...

oczywiście spełnia ten warunek, jednak zupełnie nie jest losowy!



Co to znaczy losowy?

- W analogiczny sposób można skonstruować zupełnie nielosowe ciągi liczb spełniające warunek równomierności dla dowolnej ustalonej liczby naturalnej k.
- Logiczne zatem się wydaje, że ciąg losowy powinien zawierać mniej więcej tyle samo wszystkich k-elementowych ciągów dla dowolnej liczby naturalnej k.
- Ciągi które spełniają ten warunek nazywa się nieskończenie równomiernymi (lub ∞ - równomierny)
- Czy każdy ciąg nieskończenie równomierny jest losowy?



Co to znaczy losowy?

- Załóżmy, że w rzutach kostką przyjmujemy 6=1.
- Wtedy 1 jest w uzyskanym ciągu oczywiście dwa razy więcej niż pozostałych cyfr.
- Jednak ciąg ten nadal jest właściwie losowy, ale oczywiście nie jest równomierny
- Co więcej! Można pokazać, iż istnieje cała masa ciągów nieskończenie równomiernych które nie przechodzą innych popularnych testów statystycznych losowości...
- Zatem **nieskończenie równomierny** to nie to samo co **losowy**...

Definicja losowości D. Knuth'a

- **Definicja:** Mówimy, że ciąg U_n jest losowy, gdy dla każdego efektywnego algorytmu, określającego nieskończony ciąg różnych nieujemnych liczb całkowitych s_n jako funkcję wartości U_{s0}, U_{s1},...,U_{sn}, każdy nieskończony podciąg ciągu U_{sn} utworzony przez obliczalną regułę tworzenia podciągu (przy dowolnej skończonej reprezentacji) jest równomierny[1].

Don Knuth w recepcji Open Content Aliance w San Francisco w 2005 r. Zdjęcie autorstwa Jacoba Appelbauma, dostępne na licencji Creative Commons 2.5.

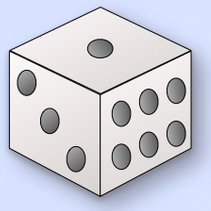
- Jak widać definicja losowości ściśle wiąże się z teorią obliczalności!
- Ciągi spełniające definicję Donalda Knuth'a istnieją, i przechodzą wszystkie popularne testy losowości (można to dowieść z definicji, bez potrzeby generowania faktycznego ciągu).

[1] Więcej szczegółów w książce D. Knuth "Sztuka Programowania", tom 2

Donald Ervin Knuth

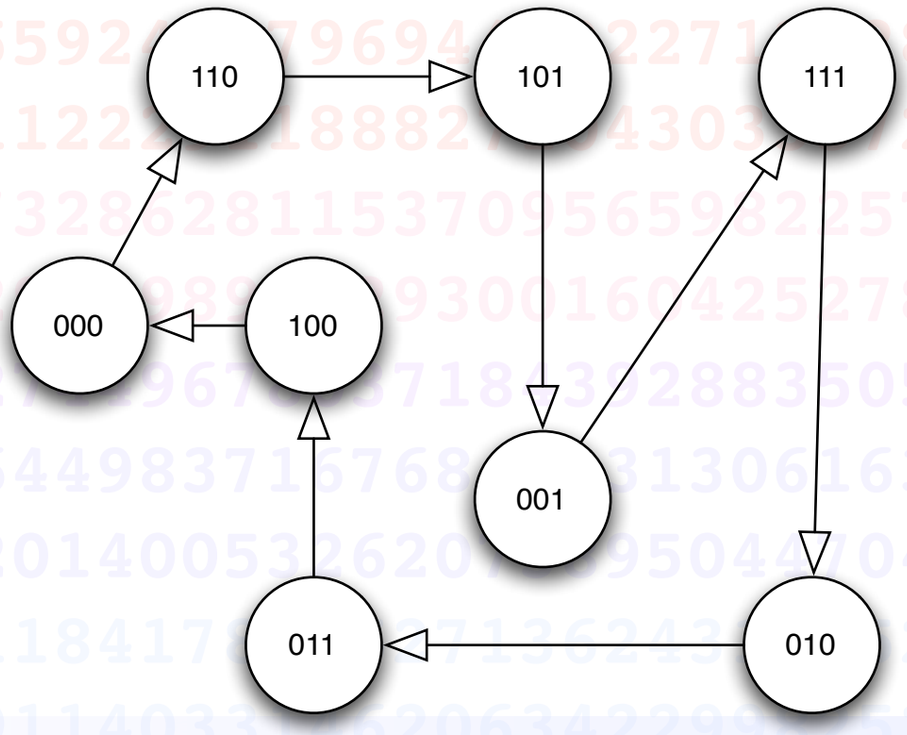
- Donald Knuth urodzony 10 stycznia 1938 roku w Milwaukee w stanie Wisconsin
- Przez wielu Knuth jest uznawany za jednego z ojców współczesnej informatyki
- Jest autorem powszechnie dzisiaj używanego systemu składu tekstu **TeX**
- Posiada spory wkład w rozwój algorytmów (np. algorytm **KMP**)
- Jest autorem wielotomowej biblii informatycznej "Sztuka programowania" uznanej za jeden z podstawowych podręczników dla studentów informatyki...

Don Knuth w recepcji Open Content Aliance w San Francisco w 2005 r. Zdjęcie autorstwa Jacoba Appelbauma, dostępne na licencji Creative Commons 2.5.

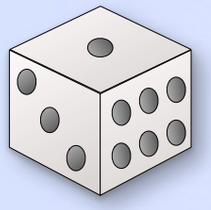


Czym jest komputer?

- Wyobraźmy sobie na chwile, że nasz komputer ma tylko 3 bity pamięci.
- Wszystkich stanów pamięci jest zatem jedynie 8.
- Z każdego stanu jest jednoznacznie określone przejście do następnego stanu.



Taki komputer to jedynie skończony graf z jedną ścieżką!

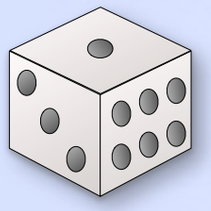


Czym jest komputer?

- W rzeczywistości każdy komputer (także ten co stoi na tym biurku) jest **automatem skończonym**, ma skończoną ilość stanów pamięci... Z każdego stanu jest jednoznaczne przejście do następnego.
- Tyle, że ilość stanów pamięci jest rzędu

$$2^{109951627776}$$

- Nawet gdyby wszystkie komputery świata zwiedzały wszystkie możliwe stany swojej pamięci z prędkością miliona na sekundę, to i tak nie zdołałyby zwiedzić wszystkich w ciągu życia wszechświata...
- Stanów jest bardzo dużo... ale **skończenie wiele**... Jaki z tego płynie wniosek?



Generowanie liczb losowych?

- Skoro komputer jest automatem skończonym, to po skończonej ilości przejść albo się **zawiesi** albo **wpadnie w cykl**!
- Zatem każdy program generujący losowe cyfry w najlepszym wypadku od pewnego miejsca **zacznie sie powtarzać**...
- Skoro tak, to ciąg cyfr w obrębie jednego okresu będzie się powtarzał w nieskończonym ciągu wygenerowanym przez ten komputer nieskończoną ilość razy...
- Taki ciąg oczywiście będzie zupełnie nierównomierny, ale i także zupełnie nielosowy...

Żaden komputer nie jest w stanie wygenerować prawdziwie losowego ciągu liczb...

Losowe, pseudolosowe ...

Ciągi losowe w ścisłym sensie

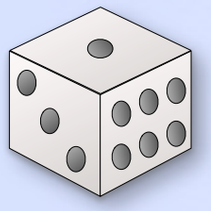
Abstrakcyjne pojęcie zdefiniowane m.in. przez Knutha. Wiadomo, że żaden komputer ich nie wygeneruje, mają właściwie znaczenie filozoficzne...

Ciągi Nieprzewidywalne

Ciągi liczb uzyskane z eksperymentów fizycznych (szumy, zakłócenia) o których nie wiemy do końca czy są losowe, ale nie potrafimy znaleźć żadnego algorytmu który by je przewidywał...

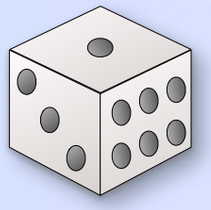
Ciągi pseudolosowe

Ciągi liczb uzyskane w wyniku działania deterministycznych algorytmów, które z pewnością nie są losowe, są jednak mocno nieuporządkowane i bez dodatkowej wiedzy sprawiają wrażenie losowych...



Liczby pseudolosowe

- Istnieją algorytmy (wzory) które na podstawie pewnych danych początkowych generują ciągi liczb, które **wyglądają zupełnie losowo**.
- Dziedzina informatyki zajmująca się generatorami pseudolosowymi stawia sobie za cel:
  - Tworzyć nowe algorytmy generowania liczb pseudolosowych
  - Badać czy generatory te są “dobre”, to znaczy ile testów statystycznych losowości przechodzą
  - Dbać o maksymalnie efektywną implementację wspomnianych algorytmów



Algorytmy generowania liczb pseudolosowych

- Metoda kongruencji liniowej
- Generator multiplikatywny
- Generator Fibonacciego
- Opóźnione generatory Fibonacciego
- Generator RC4 (Rons Code 4)
- Meresenne Twister
- Blum Blum Shub
- Generator E0
- Wiele innych...

Popularny i prosty, niekoniecznie

Modyfikacja poprzednika, trochę szybszy

Elegancki
ale bezużyteczny

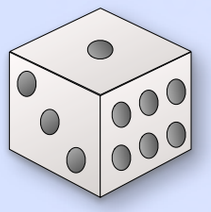
Eleganckie i wydają się dobre,
mało o nich wiadomo

Stosowany w kryptografii
(WEP), elegancki ale zbyt słaby...

Wymyślony w 1998 roku, szybki i bardzo dobry

Idealny do kryptografii ale dość wolny

Stosowany w komunikacji Bluetooth



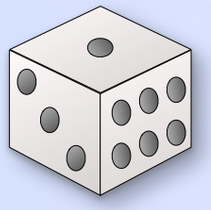
Metoda kongruencji liniowej

- Ustalamy czwórkę magicznych liczb:

$$m - \text{ moduł } (0 < m)$$
$$a - \text{ mnożnik } (0 \leq a < m)$$
$$c - \text{ krok } (0 \leq c < m)$$
$$X_0 - \text{ wartość początkowa } (0 \leq X_0 < m)$$

- Ciąg wartości pseudolosowych uzyskujemy przyjmując:

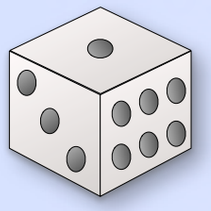
$$X_{n+1} = (aX_n + c) \bmod m$$



Jak dobrać parametry?

- Moduł powinien być możliwie duży, żeby zapewnić długi okres a jednocześnie pozwolić na szybkie obliczenia. Często wybiera się długość słowa maszyny +/- 1, czyli np. 2^{32}-1
- c powinno być względnie pierwsze z m
- a-1 powinno być wielokrotnością każdej liczby pierwszej dzielącej m
- a-1 powinno być wielokrotnością 4 jeśli m jest wielokrotnością 4.

Przykładowo: a=1664525, c=1013904223, $m=2^{32}$, $X_0=0$



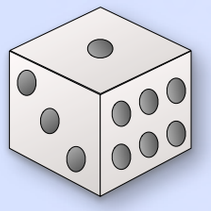
Generator multiplikatywny

- Metodę kongruencji liniowej można przyspieszyć, ustalając c=0. Uzyskamy generator postaci:

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

- Aby ten generator produkował “dobre” liczby pseudolosowe, trzeba staranniej dobrać liczby a i m.
  - m powinno być potęgą liczby pierwszej p
  - NWD(X_0,m)=1
  - a powinno być elementem pierwotnym modulo m, to znaczy wartością o największym możliwym rzędzie modulo m, gdzie rzędem nazywamy najmniejszy wykładnik spełniający:

$$a^{\lambda} \equiv 1 \bmod m$$



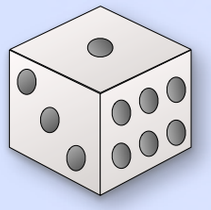
Generator Fibonacciego

- Ciekawy generator można uzyskać z eleganckiego wzoru podobnego do wzoru na liczby Fibonacciego:

$$X_n = (X_{n-1} + X_{n-2}) \bmod m$$

- Generator tej postaci nazywamy generatorem Fibonacciego
- Niestety niezależnie od wartości początkowych daje on liczby psedolosowe o **bardzo niskiej jakości**...

Zauważmy jednak, jak ładny jest ten wzór z algorytmicznego punktu widzenia! Używa się w nim jedynie jednego dodawania (które na każdym komputerze wykonuje się bardzo szybko) i jednej redukcji modulo (którą w pewnych wypadkach można zastąpić operatorem logicznym AND który również jest bardzo szybki...)



Opóźniony generator Fibonacciego

- Generator Fibonacciego można uratować stosując umiejętnie opóźnienia:

$$X_n = (X_{n-l} + X_{n-k}) \bmod m$$

$X_0, ..., X_{k-1}$ dowolne liczby całkowite, nie wszystkie parzyste

- Odpowiednio dobrana para opóźniaczy (l, k) zapewnia dobrą jakość generowanych liczb, przykładowe “dobre” opóźniacze to:

(24, 55), (38, 89), (37, 100), (30, 127), (83, 258),
(107, 378), (273, 607), (1029, 2281), (576, 3217)
(4187, 9689), (7083, 19937), (9739, 23209), ...

Generator RC4

- Algorytm RC4 jako sekwencję inicjującą przyjmuje ciąg bajtów (np. tajne hasło)
- Za jego pomocą inicjuje wewnętrzną 256 elementową permutację S[i] (Key Scheduling Algorithm) z której następnie generuje strumień pseudolosowych liczb:

KSA

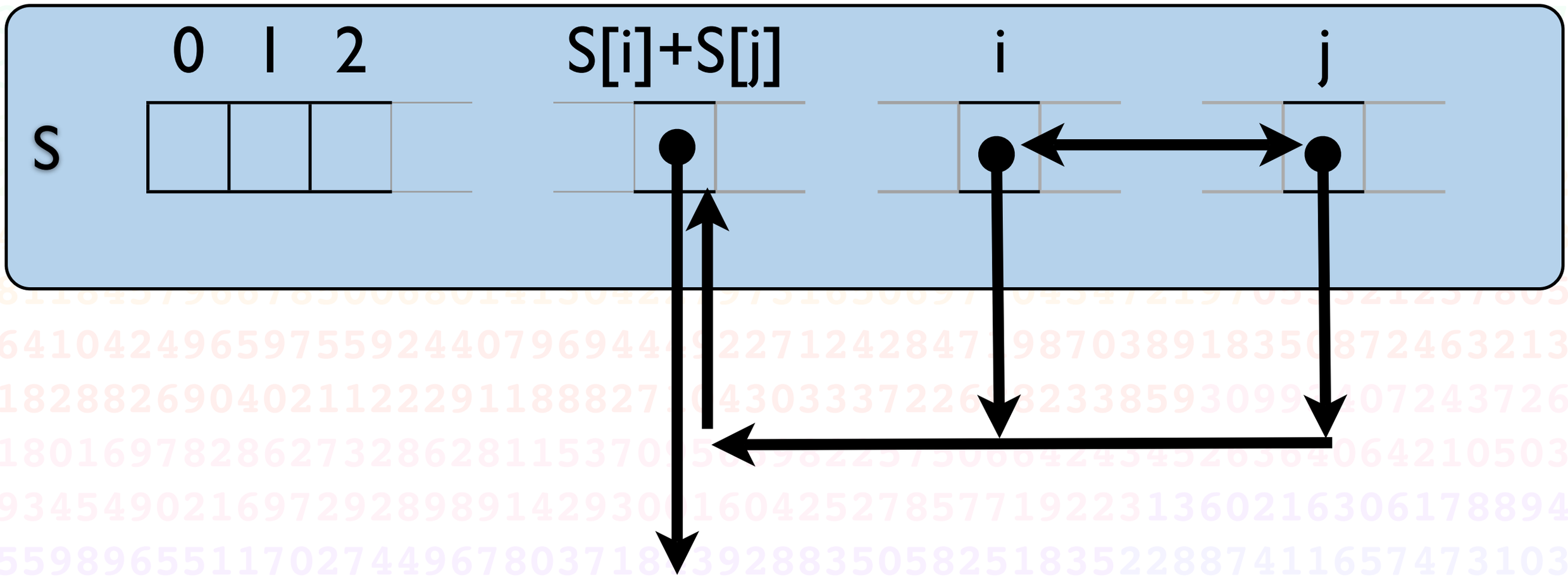
```
for i from 0 to 255
    S[i] := i
j := 0
for i from 0 to 255
    j := (j + S[i] + key[i mod keylength]) mod 256
    swap(S[i],S[j])
```

PRNG

```
i := 0
j := 0
while GeneratingOutput:
    i := (i + 1) mod 256
    j := (j + S[i]) mod 256
    swap(S[i],S[j])
    output S[(S[i] + S[j]) mod 256]
```

Generator RC4

i := (i + 1) mod 256 j := (j + S[i]) mod 256

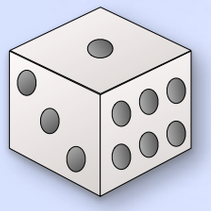


output=S[S[i]+S[j] **mod** 256]

Ronald Linn Rivest

- Ron Rivest urodzony w 1947 roku w Schenectady w stanie New York jest jednym z najsławniejszych na świecie specjalistów od kryptografii
- W 1977 roku wraz z Lenem Adlemanem i Adim Shamirem wymyślił asymetryczny kryptosystem **RSA**, dzisiaj powszechnie używany w Internecie
- Znany jest także z symetrycznych algorytmów szyfrujących znanych jako Rons Code (RC) w wersji 3,4,5 i 6.

Profesor Ron Rivest, źródło: wikipedia



Meresenne Twister

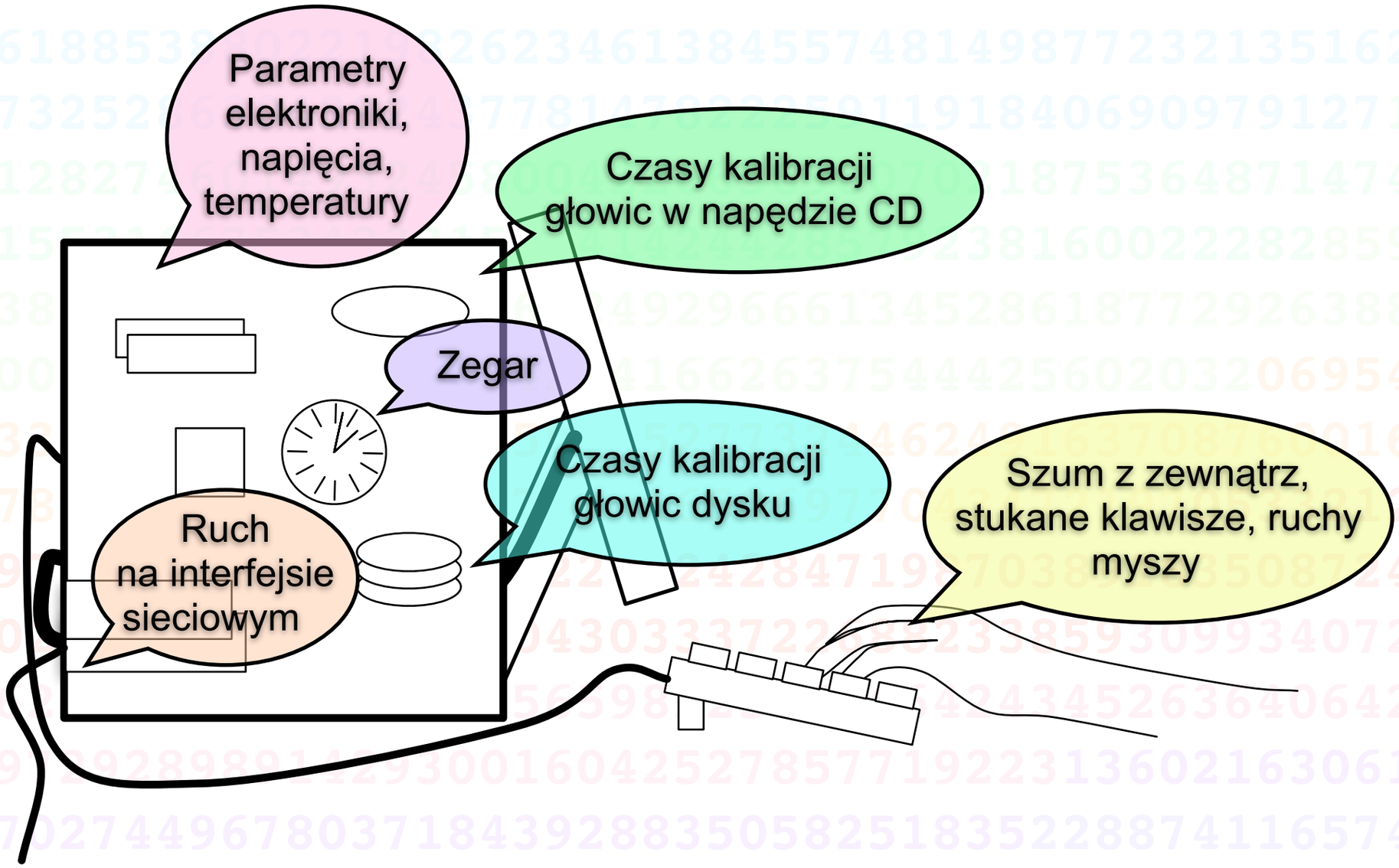
- Algorytm opracowany w 1997 roku przez Makoto Matsumoto (松本 眞) oraz Takuji Nishimura (西村 拓士) z uniwersytetu w Hiroshimie.
- Posiada kolosalny okres $2^{19937}-1$ (jest to liczba pierwsza Mersenna, stąd nazwa...)
- Nadaje się przede wszystkim do symulacji komputerowych wymagających dobrych liczb pseudolosowych (Monte Carlo), nie jest najlepszy do zastosowań w kryptografii
- Kod algorytmu nie jest tak prosty jak w przypadku metody kongruencji liniowej czy RC4, jednak w Internecie można dostać gotowe funkcje realizujące ten generator we wszystkich popularnych językach programowania...

Liczby losowe na komputerze?

- Komputer to nie tylko abstrakcyjna maszyna licząca, to także urządzenie fizyczne, które generuje sporo szumów i zakłóceń...
- Może być zatem źródłem liczb nieprzewidywalnych dla zewnętrznego obserwatora!
- Takie liczby też mają istotne zastosowanie, chociaż nikt nie zagwarantuje ich zgodności z żadnym rozkładem ani faktycznej losowości...

Nawet nieduży laptop ma w sobie mnóstwo skomplikowanych wnętrzności, które mogą stanowić źródło szumu...

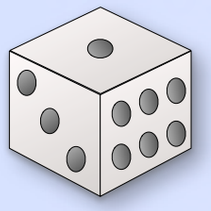
Źródła szumu w komputerze



Parametry systemu operacyjnego

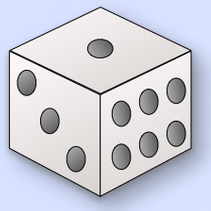
Godziny włączania/wyłączania

Zużycie zasobów



/dev/random

- Niektóre systemy operacyjne (większość Unixów, Linux, MacOSX) wykorzystują informację o szumach
- Dane są przechowywane w specjalnym (niedostępnym dla użytkownika) fragmencie pamięci zwanym **pulą entropii**
- Użytkownik może pobrać liczbę losową poprzez odczyt z pliku (urządzenia) **/dev/random**. Użytkownik dostaje funkcję skrótu z zawartości puli entropii (**SHA** - Secure Hash Algorithm)
- System dba o to aby użytkownik nie mógł poznać zawartości puli entropii aby nie mógł prognozować jej przyszłej zawartości, gwarantem tego jest funkcja **SHA** oraz blokowanie zbyt częstych odczytów...



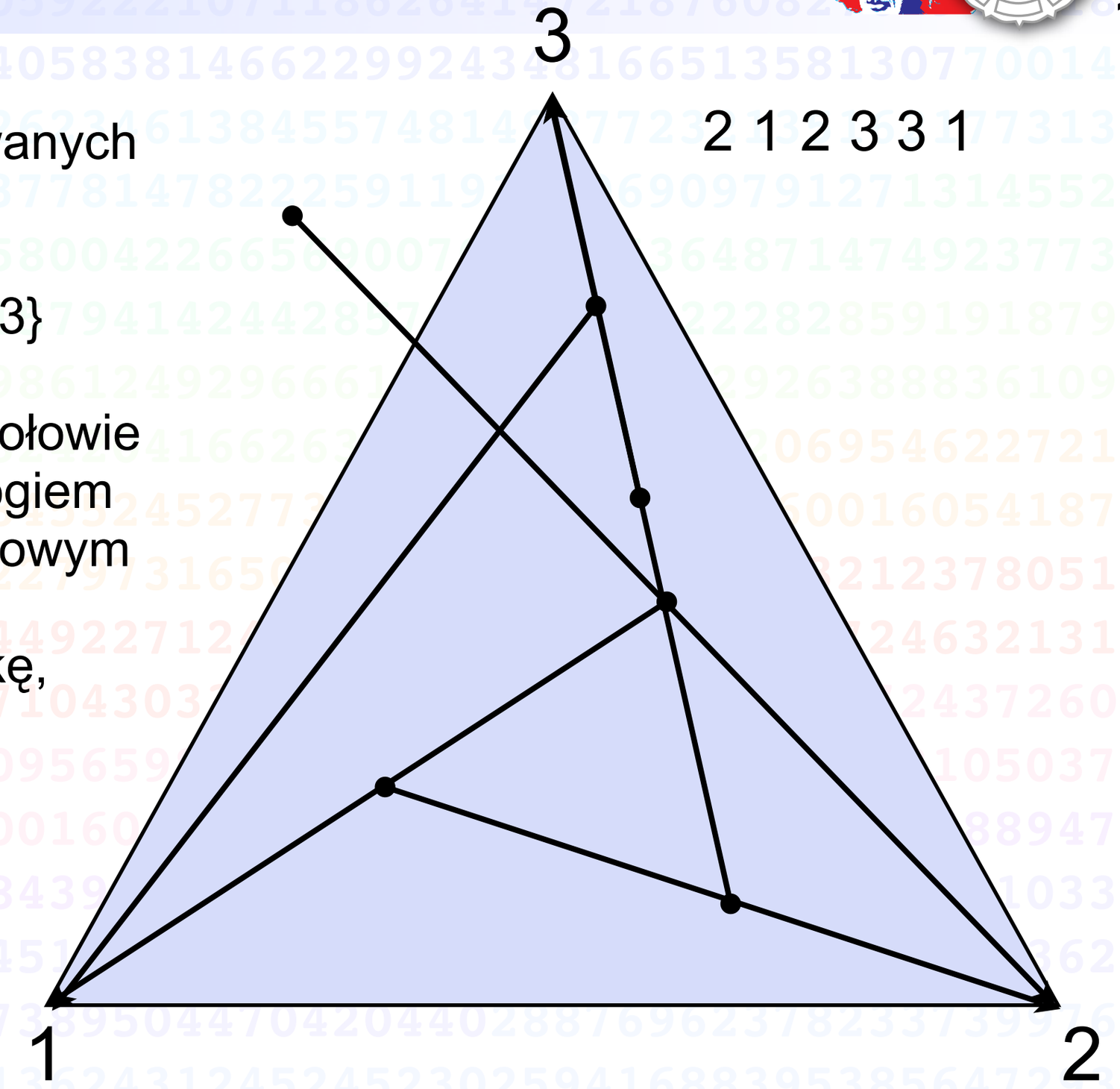
Testowanie losowości

- Test chi-kwadrat
- Test Kołmogorowa-Smirnowa
- Test równomierności
- Test odstępów
- Test spektralny
- Test autokorelacji
- Test odstępów dni urodzin
- Test pokerowy
- Test największy-z-t
- Wiele innych...

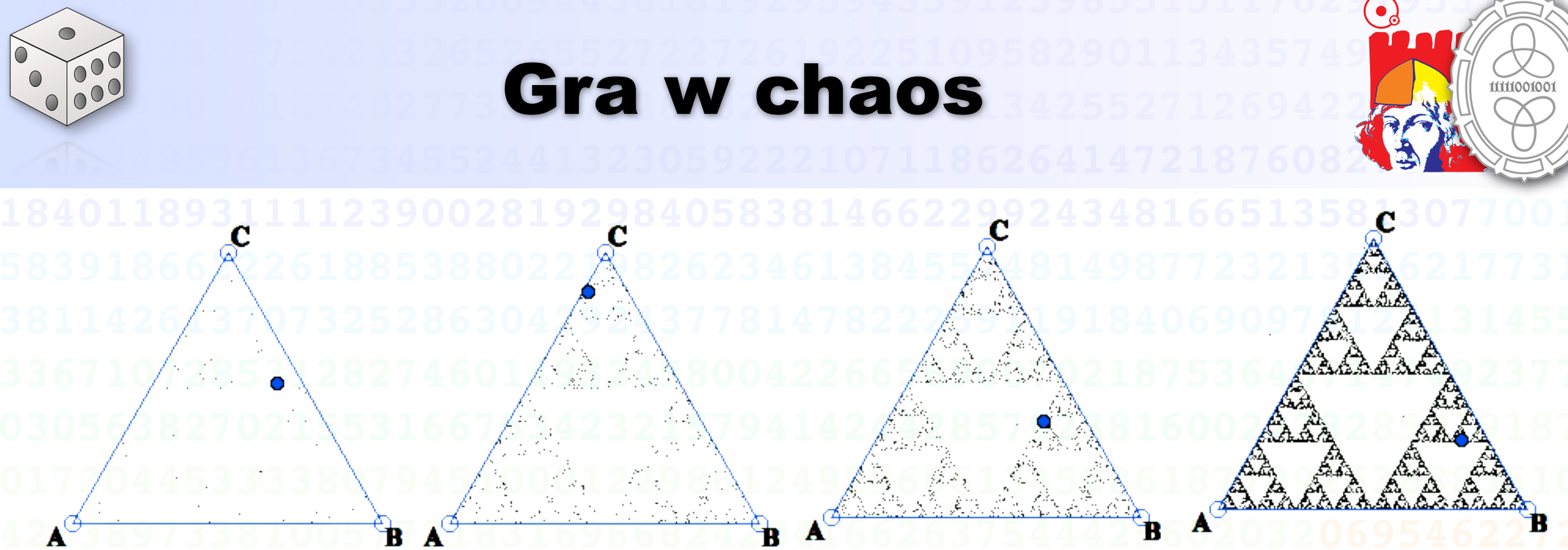
Warto przeszukać Internet w poszukiwaniu baterii testów opracowanych przez **Georgea Marsaglia**, profesora statystyki z uniwersytetu stanowego na Florydzie, znanego badacza generatorów losowych. Zestaw ten znany jest pod angielską nazwą “**Diehard tests**”...

Ciekawostka - gra w chaos

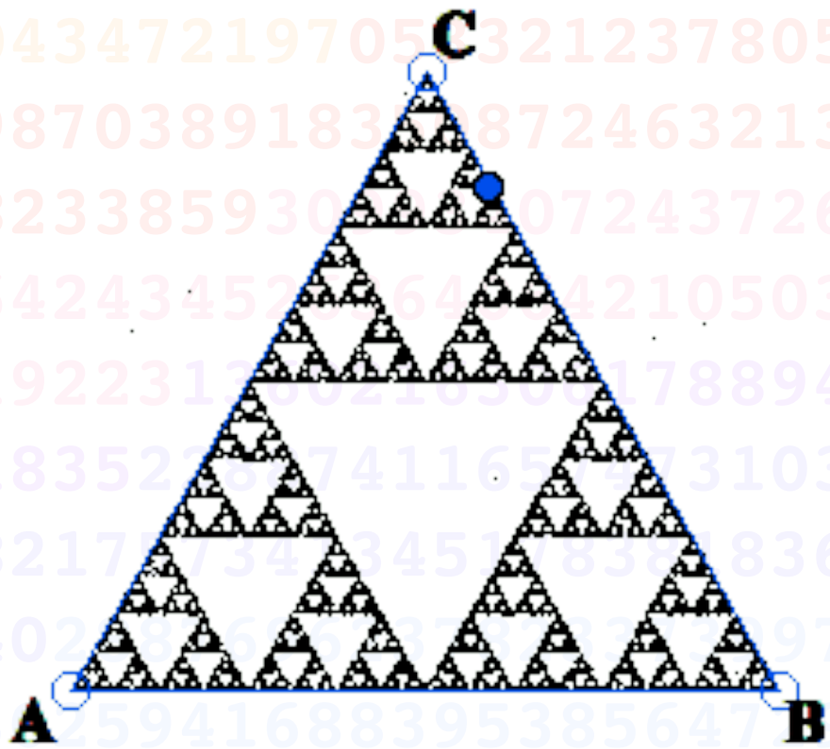
- Tworzymy trójkąt o ponumerowanych wierzchołkach i punkt startowy
- Losujemy liczbę ze zbioru {1,2,3}
- Wyznaczamy punkt będący w połowie odcinka między wylosowanym rogiem trójkąta a obecnym punktem bazowym
- Rysujemy w tym miejscu plamkę, obieramy ten punkt jako kolejny bazowy
- Wracamy do losowania
- Co powstanie gdy narysujemy dużo kropek?

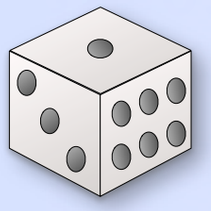


Gra w chaos



- Powstający kształt to **trójkąt Sierpińskiego**, jeden z najbardziej regularnych kształtów w matematyce!
- Co więcej! Kształt ten powstanie tylko pod warunkiem, że liczby użyte w grze będą **wystarczająco losowe** !!!

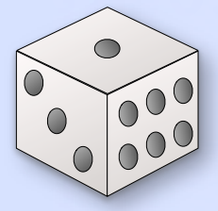




Zastosowania liczb pseudolosowych

- Liczby (pseudo)losowe znajdują wiele zastosowań w modelowaniu komputerowym (metoda Monte Carlo)
- Są potrzebne w grach komputerowych, dla urealnienia i skomplikowania zachowań przeciwników
- Są niezbędne w kryptografii, zarówno do generowania kluczy jak i blokowego szyfrowania strumieni danych
- Przydają się w grafice komputerowej do lekkiego zaburzania projektowanych detali (ludzie lubią faktury nie do końca regularne, kryjące w sobie jakąś tajemnicę)...
- Heurystyki i algorytmy stochastyczne pozwalają w przybliżony sposób rozwiązywać problemy, dla których tradycyjne algorytmy mają zbyt wielką złożoność

Hierarchia problemów w informatyce



Nierozstrzygalne

Rozstrzygalne

Złożoność wyrażalna funkcją elementarną

Złożoność wykładnicza

Pamięć wielomianowa
P -Space

NP ?

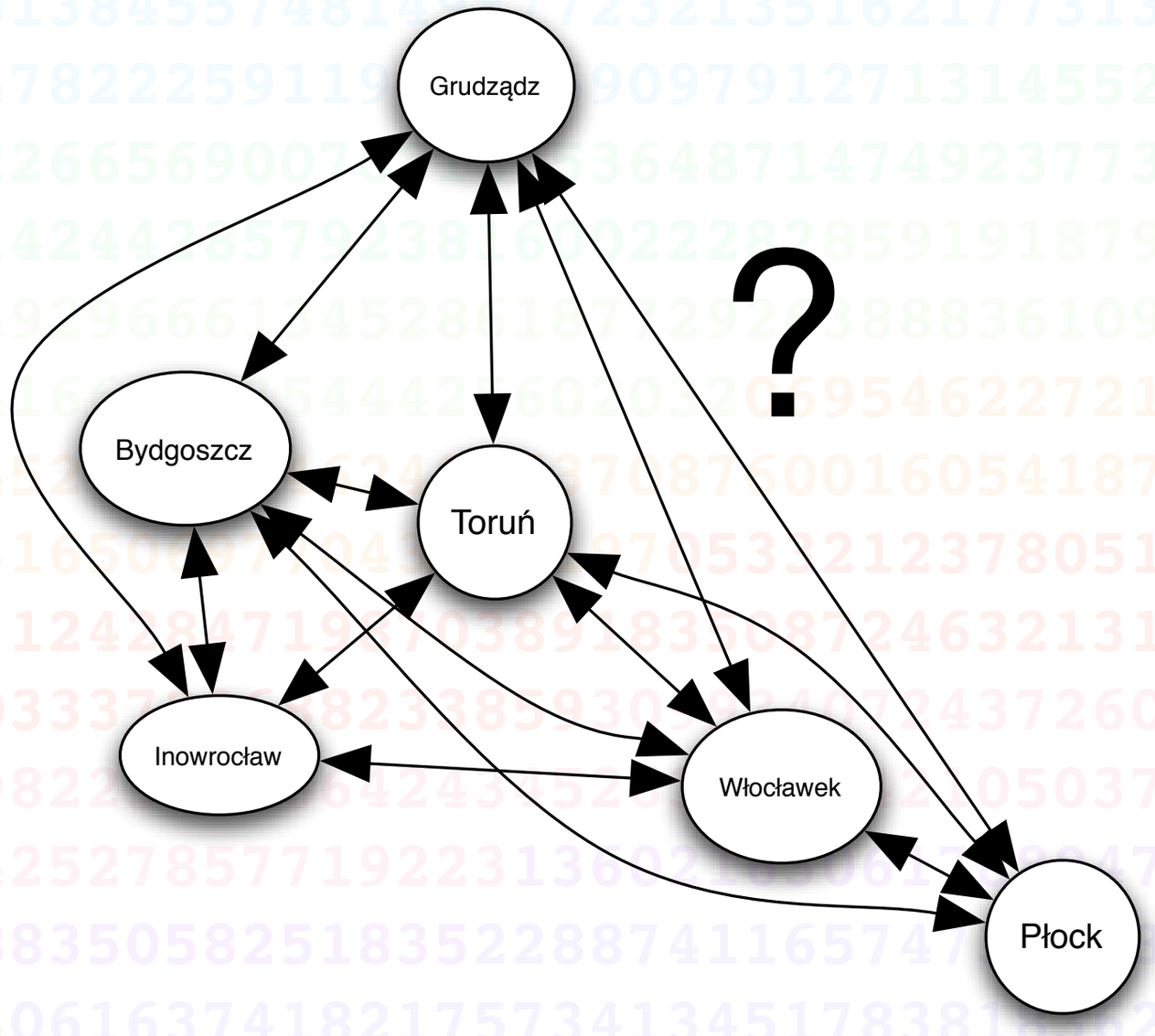
Złożoność wielomianowa
P

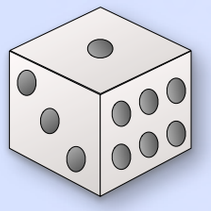
W praktyce potrafimy rozwiązać jedynie problemy o złożoności wielomianowej (zarówno pamięciowej jak i czasowej)...

Pozostałe musimy rozwiązywać metodami przybliżonymi

Przykład trudnego problemu

- Mamy zbiór miast oraz odległości między nimi
- Zaplanować trasę podróży komiwojażera, by każde miasto odwiedził dokładanie raz i pokonał minimalną odległość
- Problem ten nosi nazwę **problemu komiwojażera** i jest klasycznym przykładem tak zwanego problemu **NP-zupełnego**





Jak zaprząc losowość do rozwiązywania problemów?

- Jeśli nie ma innego wyjścia, możemy wybierać losowo rozwiązanie i sprawdzać czy jest dobre...
- Wymaga to jednak posiadania szybko obliczalnej miary oceniającej jakość rozwiązania...
- Algorytmy tego typu często nazywa się algorytmami **random walk** (losowy spacer), jako że zupełnie losowo przeglądają przestrzeń rozwiązań...
- Przykład. wybieramy losowo 10000 razy drogę komiwojażera, jako rozwiązania zwracamy tą, która oferowała najmniejszą sumaryczną odległość...

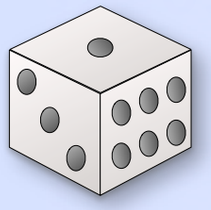
Niestety to bardzo nieefektywny sposób programowania, zazwyczaj mamy do dyspozycji pewną dodatkową wiedzę o własnościach rozwiązania, którą możemy wykorzystać...

Symulowane wyżarzanie

- Regularne kryształy są takim ułożeniem atomów w którym minimalizowana jest energia wynikająca w wzajemnych oddziaływań
- Jak wyhodować monokryształ? Trzeba rozgrzać substancję do stanu ciekłego a następnie powoli schładzać...
- Jeśli schłodzimy zbyt szybko powstanie zlepek małych kryształków, wda się pewien nieporządek którego chcemy uniknąć...
- Takie postępowanie nazywa się wyżarzaniem...
- **Ale co to ma wspólnego z informatyką?**

Kryształ kwarcu

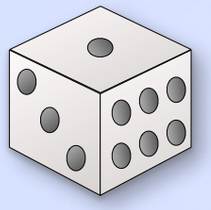
Jeden z największych na świecie sztucznie wyhodowanych monokryształów (Saint-Gobain)



Symulowane wyżarzanie

A gdyby:

- Tak skonstruować hipotetyczny układ fizyczny, by jego **stan** modelował **rozwiązanie** pewnego problemu a jego **energia** **jakość** tego rozwiązania?
- Wtedy wystarczy zainicjować go losowo, a następnie symulować wyżarzanie, tak aby po odpowiednim obniżeniu temperatury “skrystalizował” w stanie o najniższej energii (najlepszym rozwiązaniu)?
- Ten sposób postępowania to **symulowane wyżarzanie**
- Jak się okazuje, pojęcie **temperatury** ma wiele wspólnego z **losowością**, obniżanie temperatury to nic innego jak zmniejszanie czynnika losowego w algorytmie!

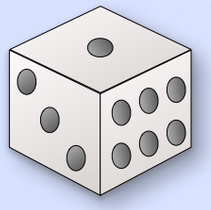


Algorytmy Monte-Carlo

- Czasem chcemy obliczyć nie dokładną a jedynie **średnią** wartość parametru na pewnym zbiorze...
- Możemy wtedy skorzystać z własności:

$$\sum_{i=1}^{n} \frac{f(X_i^{(p)})}{n} \approx \int f(x)dp(x)$$

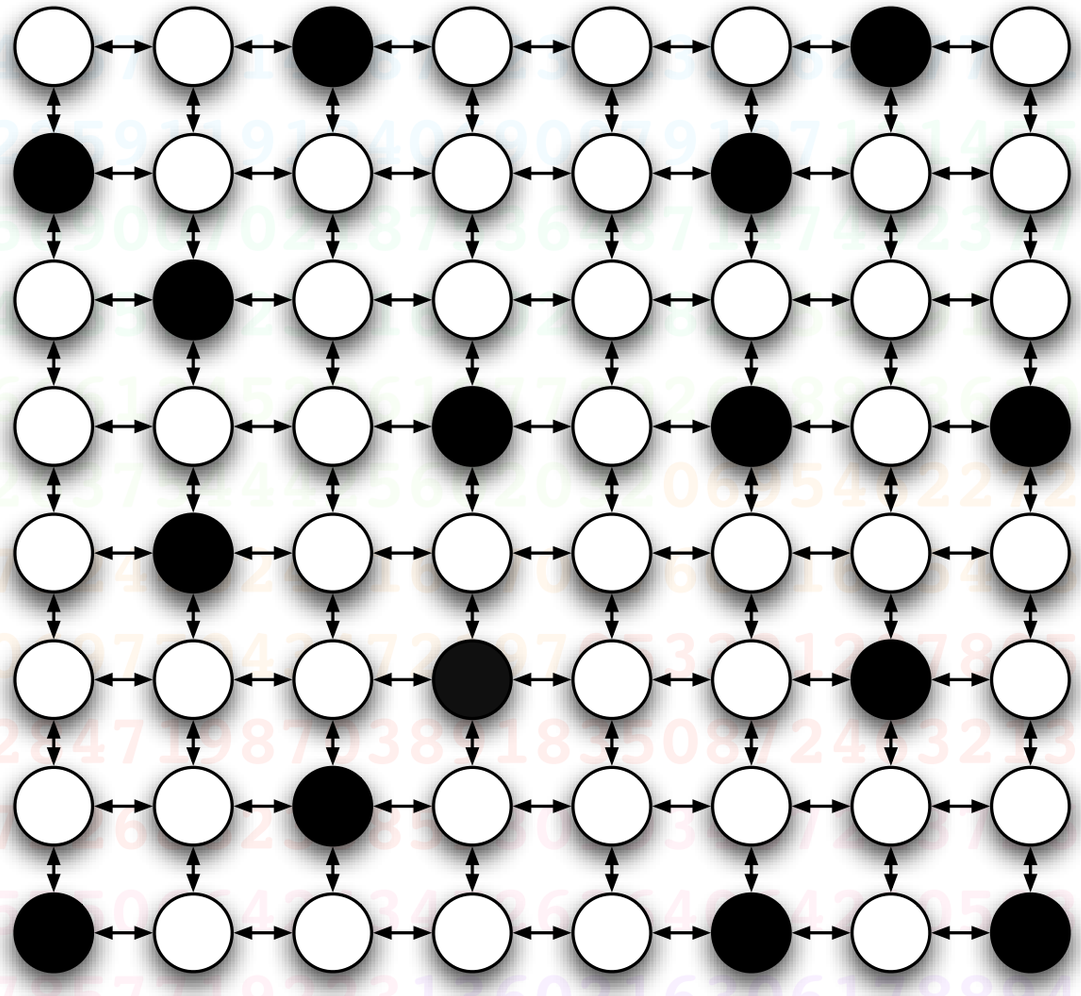
- Oznacza to, że całkę z funkcji względem rozkładu możemy przybliżać poprzez średnią wartości funkcji w punktach wylosowanych z tego rozkładu...
- Metoda Monte-Carlo jest powszechnie stosowana w modelowaniu skomplikowanych zjawisk fizycznych itp...



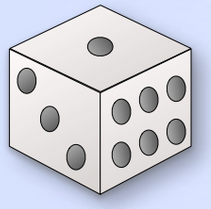
Markov Chain Monte Carlo

- “wylosować z rozkładu” brzmi trywialnie... ale często jest to główna trudność w wykorzystaniu metody Monte-Carlo, rozkłady często mogą być bardzo skomplikowane...
- Przykład: mamy kratę jak obok, w której żaden czarny węzeł nie ma czarnego sąsiada. Jak przeprowadzić losowanie wśród krat o tej własności **żeby każda z nich była jednakowo prawdopodobna**?

  Ilość wszystkich konfiguracji już dla przypadku 8x8 jest 2^{64}, tych dozwolonych jest zapewne również astronomicznie wiele...



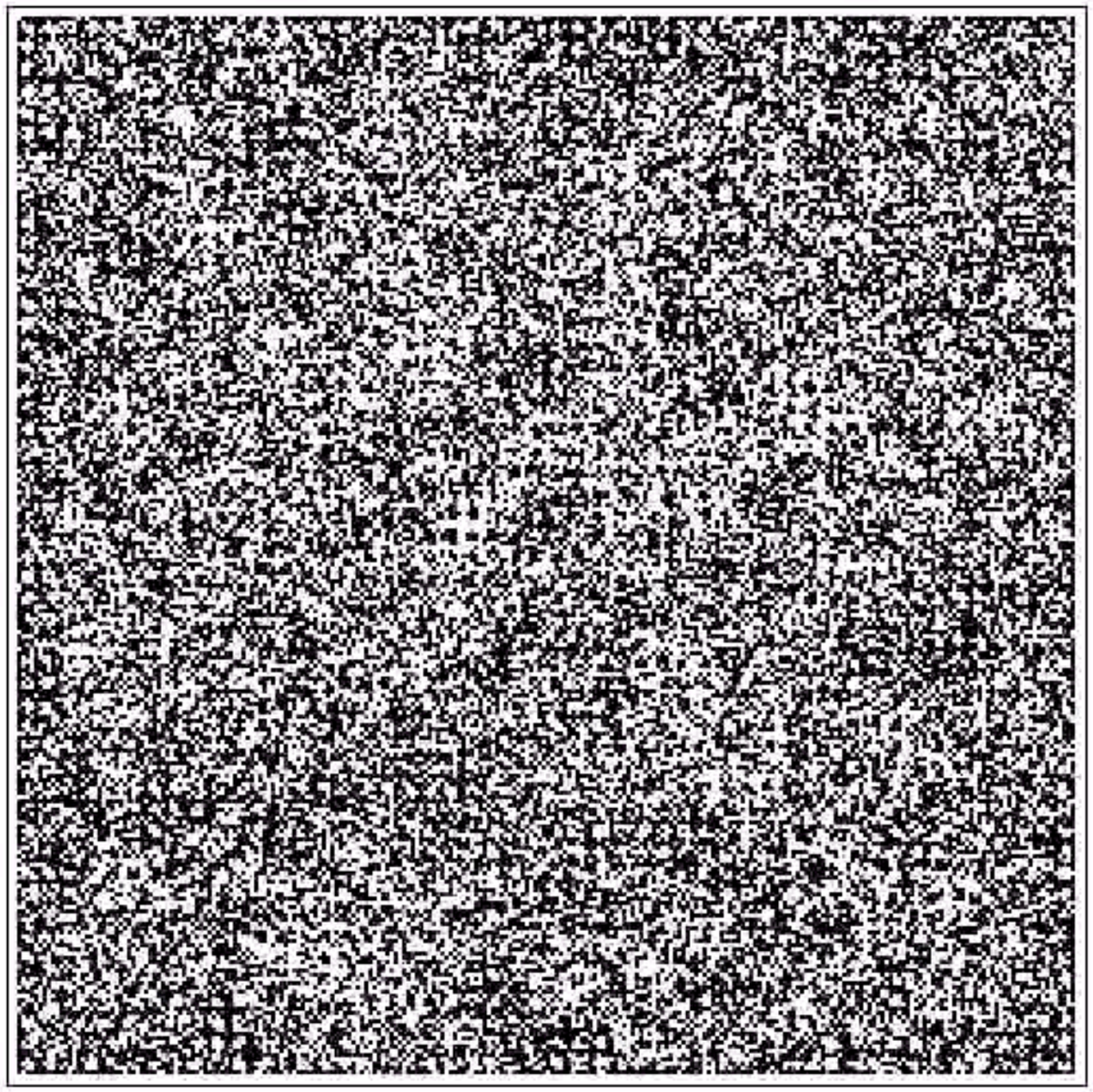
Losowa krata w której żadna krawędź nie łączy dwóch czarnych wierzchołków.



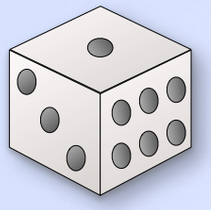
Markov Chain Monte Carlo

- Aby wylosować zmienne z tak skomplikowanych rozkładów trzeba zaprząc zaawansowaną **teorię procesów stochastycznych**...
- Da się to zrobić poprzez odpowiednie skonstruowanie specjalnego procesu losowego (procesu Markova), który w miarę symulowania osiąga tak zwany **rozkład stacjonarny**...
- Jeśli skonstruujemy łańcuch tak aby jego rozkład stacjonarny był tym samym rozkładem w którego chcemy losować, to poprzez wielokrotne zasymulowanie łańcucha możemy uzyskać wymaganą próbkę losową...
- Jednak czasem już samo symulowanie łańcucha kryje **niebanalne zjawiska**...

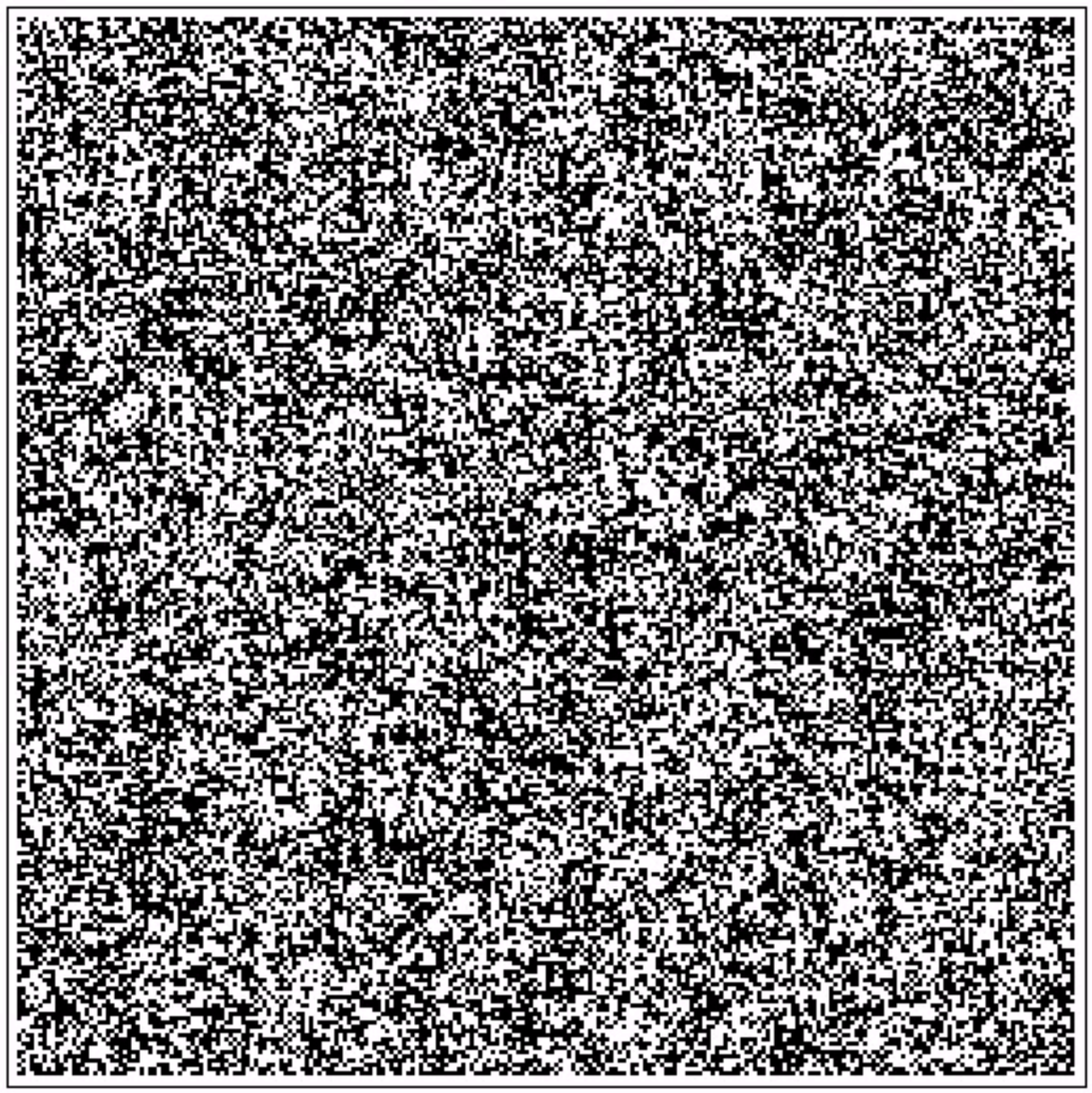
Markov Chain Monte Carlo



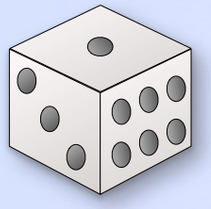
Ewolucja specyficznego łańcucha Markova w wysokiej temperaturze (duży nieporządek)



Markov Chain Monte Carlo



Ewolucja specyficznego łańcucha Markova w niskiej temperaturze (spora regularność zachowania)

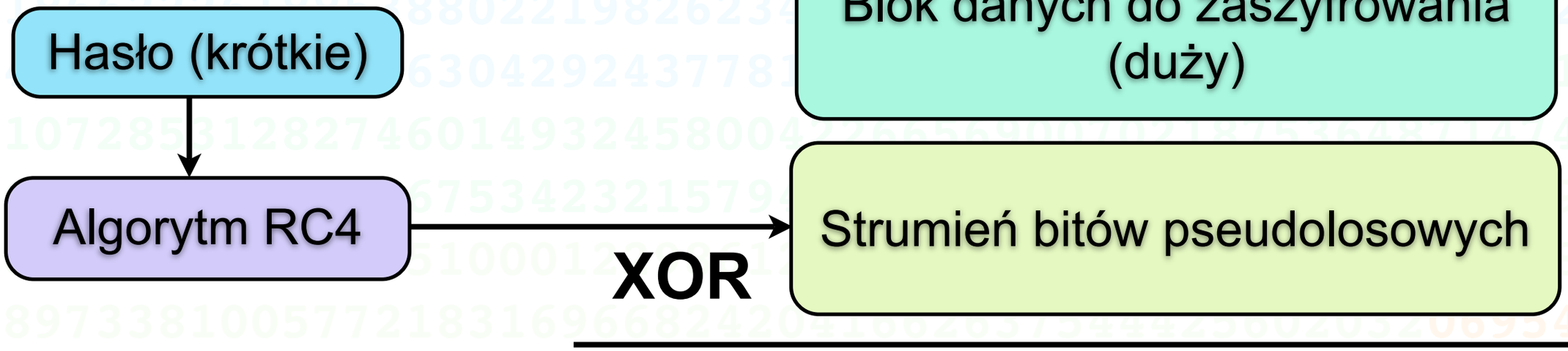


Zastosowania liczb losowych w kryptografii

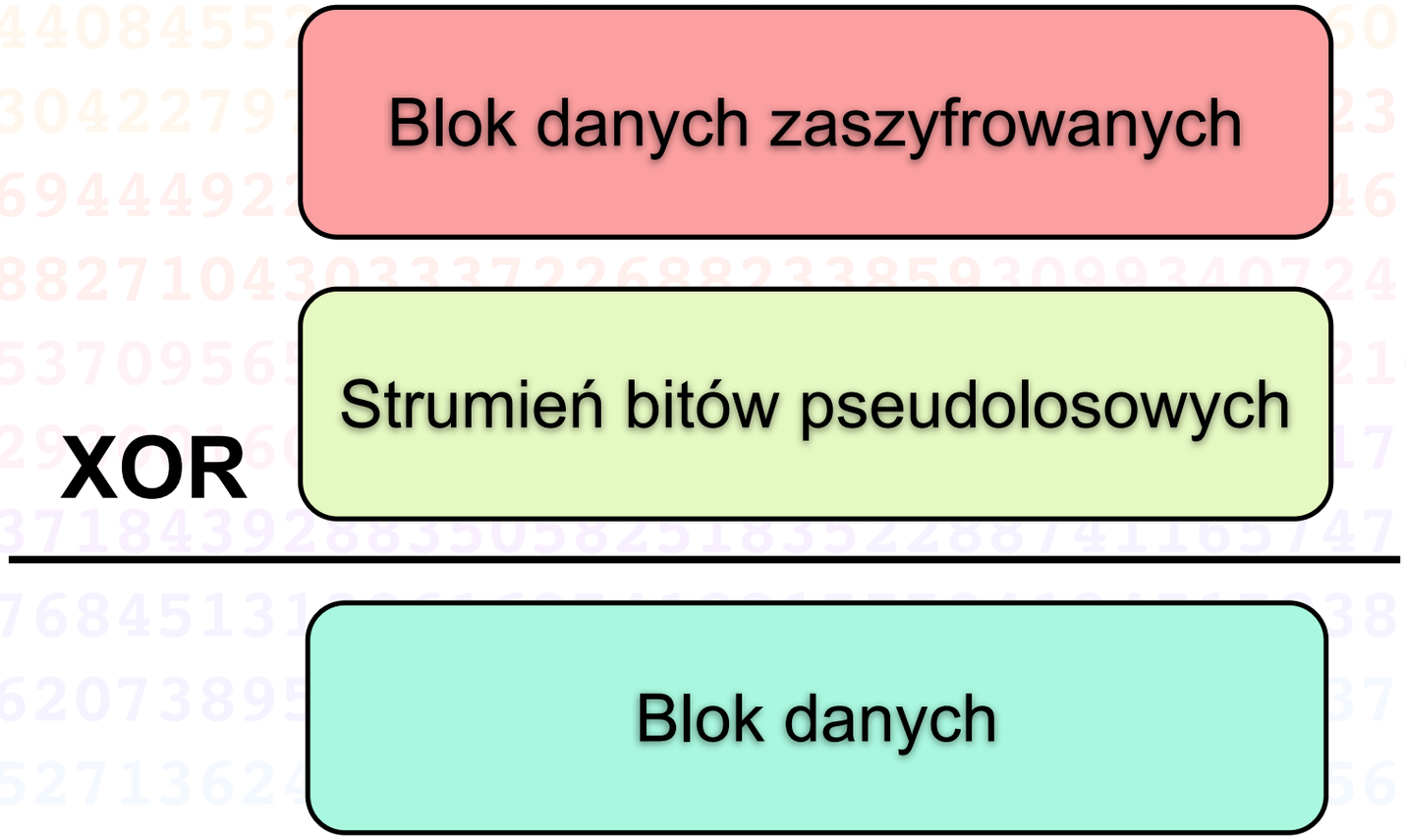
- Prominentnym zastosowaniem wszelkiej losowości jest kryptografia.
- Programy szyfrujące muszą zagwarantować unikatowość (w skali całego świata) kluczy generowanych na komputerze... w tym celu chętnie korzystają np. z /dev/random
- Także specyficzne generatory liczb pseudolosowych znajdują zastosowania do generowania strumieni szyfrujących długie porcje danych...

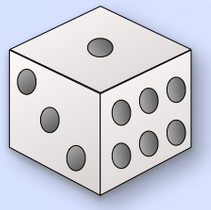
Klucz cyfrowy różni się dramatycznie od tradycyjnego kawałka metalu. To po prostu ciąg bitów o pewnych ściśle określonych własnościach. Bezpieczne generowanie tych kluczy wymaga źródła liczb losowych nieprzewidywalnych dla potencjalnych wrogów...

Szyfrowanie RC4



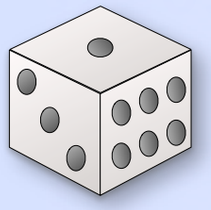
XOR	0	1
0	0	1
1	1	0





Podsumowanie

- Wygenerowanie czegokolwiek losowego na komputerze nie jest wcale proste!
- Liczby losowe znajdują jednak masę zastosowań, począwszy od kryptografii a na grach komputerowych skończywszy...
- **“Losowe”** to nie to samo co “**pseudolosowe”**, to z kolei nie to samo co “**nieprzewidywalne”**...
- Tworzenie algorytmów generowania liczb pseudolosowych to spory kawałek informatyki...



Pytania

Prezentacja dostępna jest pod adresem:

http://www.mat.uni.torun.pl/~philip/fnis2006.pdf