

Generowanie liczb pseudolosowych

Najprostszym sposobem generowania liczb pseudolosowych jest wykorzystanie funkcji `rand()`, wywodzącej się jeszcze z języka C. Problemem może być jednak zbyt mały zakres generowanych liczb. Częstym błędem jest przyjęcia założenia, że funkcja `rand()` daje maksymalną wartość taką samą, jak dla liczby typu `int`. W praktyce zwykle, nawet na platformie 64-bitowej, maksymalna wartość jest równa 32767. Można się o tym przekonać wyświetlając wartość stałej `RAND_MAX`.

Zbyt mały zakres liczb pseudolosowych może być problemem przy wykonywaniu testów dla dużych struktur danych, na przykład tablicy o rozmiarze 1 000 000. Konieczne może być wtedy wygenerowanie losowego indeksu tablicy w zakresie 0 .. 999 999.

Aby uzyskać liczby losowe w zakresie większym niż zwracane przez funkcję `rand()` można próbować tworzyć większe liczby z dwóch kolejno wylosowanych liczb lub użyć generatora liczb pseudolosowych z biblioteki `<random>`, opis można znaleźć w poniższych linkach:

<https://msdn.microsoft.com/en-us/library/bb982398.aspx>

<http://en.cppreference.com/w/cpp/numeric/random>

W pokazanym poniżej przykładzie generowane są liczby z zakresu 1 .. 1 000 000, określana jest również uzyskana wartość minimalna i maksymalna. Przy odpowiednio dużej liczbie testów (na przykład `TESTS = 10 000 000`) można zauważyć, że udaje się uzyskać skrajne wartości z zadanego zakresu (czyli wartość minimalną 1 i odpowiednio maksymalną 1 000 000).

```
#include <iostream>
#include <random>

using namespace std;

#define TESTS 10000000

//-----
int main()
{
    int val, min, max;
    random_device rd; // non-deterministic generator
    mt19937 gen(rd()); // random engine seeded with rd()
    uniform_int_distribution<> dist(1, 1000000); // distribute results between
                                                // 1 and 1000000 inclusive

    min = INT_MAX;
    max = 0;

    for (int i = 0; i < TESTS; i++)
    {
        val = dist(gen); // pass the generator to the distribution

        if(val < min)
            min = val;

        if(val > max)
            max = val;
    }

    cout << "min = " << min << " max = " << max << endl;

    system("PAUSE");
    return(0);
}
```

W przedstawionym przykładzie maszyna losująca (*random engine*) czyli *gen()* (w tym przypadku 32-bit Mersenne twister engine) inicjowana jest liczbą losową, uzyskaną z niedeterministycznego generatora *rd()*. Jeśli chcemy uzyskać powtarzalne rezultaty przy każdym uruchomieniu programu do zainicjowania można użyć stałej wartości.

Wartości losowe, uzyskane z maszyny losującej, przekazywane są jako argument do odpowiedniego rozkładu *dist()* w zadanym zakresie. W przykładzie jest to rozkład równomierny ale mechanizm ten umożliwia również uzyskanie wielu innych rozkładów (na przykład Gaussa).

Maszyna losująca *mt19937* (Mersenne twister, Matsumoto i Nishimura, 1998) jest jedną z najszybszych. Stosując inne trzeba się zwykle liczyć z dłuższym czasem generowania wartości losowych (w skrajnych przypadkach nawet kilkunastokrotnie).