TASODIFIY SONLARGA OID ALGORITMLAR

АЛГОРИТМЫ СЛУЧАЙНЫХ ЧИСЕЛ

RANDOM NUMBER ALGORITHMS

Abdullayev Shaxboz Solijon o‘g‘li
FarDU Axborot texnologiyalari kafedrasi katta o'qituvchisi
shaxbozfardu2023@gmail.com
ORCID ID 0000-0001-9382-732X
Xolmatova Gulhayo Hasan qizi
FarDU Axborot tizimlari va texnologiyalari yo’nalishi 1-kurs talabasi
xolmatovagulhayo1@gmail.com

Annotatsiya. *Tasodifiy sonlarga oid algoritmlar (Random Number Algorithms) - tasodifiy sonlarni ishlab chiqarish, ularning sifatini baholash va qo'llanilishini ta'minlash bo'yicha ilmiy va amaliy muammolarni o'rganadi. Ushbu maqolada tasodifiy sonlar va ular yaratishning asosiy usullari, xususan, psevdotasodifiy sonlar yaratishda qo'llaniladigan algoritmlar, masalan, Liner Kongruensial Generator (LCG) va Mersenne Twister kabi algoritmlar haqida so'z yuritiladi. Shuningdek, tasodifiy sonlarning amaliy sohalardagi qo'llanilishi, xususan, kriptografiya, simulyatsiya va o'yinlar kabi sohalarda qanday ishlatilishi va tasodifiy sonlar sifatini tekshirish metodlari keltiriladi. Maqola tasodifiy sonlarga oid algoritmlarning samaradorligi va xavfsizligini baholashga qaratilgan zamonaviy tadqiqotlar va amaliy qo'llanmalar haqida ham ma'lumot beradi*

Аннотация. *Алгоритмы случайных чисел — это область исследования, посвященная созданию случайных чисел, оценке их качества и обеспечению их применения. В данной статье рассматриваются основные методы генерации случайных чисел, в частности алгоритмы, используемые для создания псевдослучайных чисел, такие как Линейный конгруэнц-генератор (LCG) и Мерсенн-Твистер. Также обсуждается использование случайных чисел в практических областях, таких как криптография, моделирование и игры, а также методы проверки качества случайных чисел. Статья также*

предоставляет информацию о современных исследованиях и практическом применении алгоритмов случайных чисел для оценки их эффективности и безопасности.

Abstract. *Random number algorithms are an area of research focused on generating random numbers, evaluating their quality, and ensuring their application. This article discusses the primary methods for generating random numbers, particularly algorithms used for generating pseudorandom numbers, such as the Linear Congruential Generator (LCG) and Mersenne Twister. It also explores the use of random numbers in practical fields like cryptography, simulation, and gaming, as well as methods for testing the quality of random numbers. The article also provides insights into modern research and practical applications of random number algorithms for assessing their efficiency and security.*

Kalit so‘zlar: *Tasodifiy sonlar, Algoritmlar, Psevdotasodifiy sonlar, Kriptografiya, Liner kongruensial generator, Mersenne, Twister, Simulyatsiya, Tasodifiylik testi, Monte-Karlo metodlari, Xavfsizlik*

Ключевые слова: *случайные числа, алгоритмы, псевдослучайные числа, криптография, линейный конгруэнциальный генератор, мерсенн, твистер, симуляция, тест на случайность, методы монте-карло, безопасность*

Keywords: *random numbers, algorithms, pseudorandom numbers, cryptography, linear congruential generator, mersenne, twister, simulation, randomness test, monte-carlo methods, security*

KIRISH

Tasodifiylik tushunchasi qadim zamonlardan buyon insoniyat e’tiborini tortib kelgan. Falsafa, ehtimollar nazariyasi, matematika va fizika sohalarida tasodifiy hodisalarni o‘rganish orqali insoniyat tabiat qonunlarini yanada chuqurroq tushunishga harakat qilgan. Bugungi raqamli asrda esa tasodifiylik tushunchasi nafaqat nazariy tadqiqotlarda, balki real amaliyotda ham muhim ahamiyat kasb etadi. Ayniqsa, kompyuter texnologiyalarining jadal rivojlanishi tasodifiy sonlarga oid algoritmlarga bo‘lgan talabni keskin oshirdi.

Kompyuter tizimlarida tasodifiy sonlar ko‘plab yo‘nalishlarda qo‘llaniladi. Masalan, statistik modellashtirishda, Monte-Karlo usullarida, kriptografik tizimlarda maxfiylikni ta’minlashda, algoritmik o‘yin dizaynida realistik harakatlarni

simulyatsiya qilishda, sun’iy intellektda trening ma’lumotlarini tasodifiy aralashtirishda va hatto san’atda generativ tasvirlar yaratishda ham tasodifiylik asosiy vosita sifatida xizmat qiladi.

Shuni alohida ta’kidlash kerakki, zamonaviy kompyuterlar deterministik — ya’ni oldindan aniq belgilangan qoidalarga asoslangan tizimlar bo‘lib, ular o‘z-o‘zidan “haqiqiy” tasodifiy sonlarni hosil qila olmaydi. Shu sababli, dasturchilar va olimlar maxsus algoritmlar orqali **psevdo-tasodifiy sonlar**ni generatsiya qilish usullarini ishlab chiqishgan. Bu algoritmlar ma’lum boshlang‘ich qiymat — *urug‘ (seed)* asosida sonlar ketma-ketligini hosil qiladi va bu ketma-ketlik statistik jihatdan tasodifiyga juda yaqin bo‘ladi. Turli algoritmlar turli darajadagi sifat va samaradorlikni ta’minlaydi. Ba’zilarida tezlik ustunlik qilsa, boshqalarida xavfsizlik va nazorat qilish imkoniyati muhimroq bo‘ladi.

Tasodifiy sonlar algoritmlari ikki katta toifaga bo‘linadi: Psevdo-tasodifiy sonlar generatorlari (PTSG) va Haqiqiy tasodifiy sonlar generatorlari (HTSG).

Psevdo-tasodifiy sonlar generatorlari (PTSG) – matematik formulalar orqali oldindan belgilangan algoritm asosida ishlaydi. Masalan, Linear Congruential Generator (LCG), Mersenne Twister, Xorshift va boshqalar.

Haqiqiy tasodifiy sonlar generatorlari (HTSG) – fizik hodisalarga, masalan, radioaktiv parchalanish, issiqlik shovqini yoki kvant effektlarga asoslanadi. Ular maxsus apparat vositalari yordamida amalga oshiriladi va sof tasodifiylikni ta’minlaydi.

Psevdo-tasodifiy sonlar real tizimlarda ko‘p hollarda yetarli bo‘lsa-da, xavfsizlik talab qilinadigan sohalarda, ayniqsa kriptografiyada, haqiqiy tasodifiylikka yaqin bo‘lgan algoritmlar talab etiladi. Shuning uchun zamonaviy PTSG’lar ko‘pincha yuqori darajadagi statistik testlardan o‘tkaziladi: masalan, Diehard testlar to‘plami yoki NIST tomonidan ishlab chiqilgan testlar yordamida ularning sifati baholanadi.

Dasturning kodi :

```
#include <iostream>
#include <random>
#include <ctime>
using namespace std;
int main() {
    int sonlar_soni;
```

```
cout << "Nechta tasodifiy son generatsiya qilishni xohlaysiz? ";
cin >> sonlar_soni;
cout << endl << "1. Klassik rand() funksiyasi yordamida:" << endl;
srand(time(0));
for (int i = 0; i < sonlar_soni; ++i) {
   int tasodifiy_son = rand() % 100;
   cout << tasodifiy_son << " ";   }
cout << endl;
cout << endl << "2. Mersenne Twister (mt19937) yordamida:" << endl;
mt19937 generator(static_cast<unsigned int>(time(0)));
uniform_int_distribution<int> butun_oraliq(0, 99);
for (int i = 0; i < sonlar_soni; ++i) {
   int tasodifiy_son = butun_oraliq(generator);
   cout << tasodifiy_son << " "; }
cout << endl;
cout << endl << "3. Haqiqiy (real) sonlar [0.0, 1.0) oralig'ida:" << endl;
uniform_real_distribution<double> haqiqiy_oraliq(0.0, 1.0);
for (int i = 0; i < sonlar_soni; ++i) {
   double haqiqiy_son = haqiqiy_oraliq(generator);
   cout << haqiqiy_son << " "; }
cout << endl;
return 0;  }
```

Dastur tahlili

#include <iostream> - Bu kutubxona fayli konsolga ma'lumot chiqarish (cout) va konsoldan ma'lumot olish (cin) imkoniyatlarini beradi. Bu satr dasturda ishlatilgan barcha konsol bilan aloqador funktsiyalarni chaqirishi uchun zarur.

#include <string> - Bu kutubxona fayli string turini ishlatishga imkon beradi. string o'zgaruvchisi matnli ma'lumotlarni saqlash uchun ishlatiladi.

#include <random> - bu kutubxona zamonaviy tasodifiy sonlar generatorlari (masalan, mt19937) va ularning taqsimotlarini (uniform_int_distribution, uniform_real_distribution) ishlatish uchun kerak.

#include <ctime> - time(0) funksiyasini ishlatish uchun kerak. Bu funksiyadan urug‘ (seed) yaratishda foydalaniladi, ya’ni vaqtga asoslangan boshlang‘ich qiymat.

using namespace std; - Bu satr std nomli kutubxonani dasturga kiritadi. std namespace ichidagi barcha elementlardan bevosita foydalanish imkonini beradi. Masalan, cout, cin, string kabi elementlar to'g'ridan-to'g'ri chaqiriladi va "std::" prefiksini qo'llash kerak emas.

int main() - Dastur bajarilishi shu funksiyadan boshlanadi.

int sonlar_soni - Foydalanuvchi nechta tasodifiy son kiritmoqchi ekanligini saqlash uchun butun sonli (int) o‘zgaruvchi.

cout<<"Nechta tasodifiy son generatsiya qilishni xohlaysiz?" - Foydalanuvchidan sonlar sonini so‘raydi.

cin >> sonlar_soni; - Klaviaturadan foydalanuvchi kiritgan qiymatni sonlar_soni o‘zgaruvchisiga yuklaydi.

cout << endl << "1. Klassik rand() funksiyasi yordamida:" << endl; - Bo‘lim 1: Klassik rand() usuli bilan son generatsiyasini boshlayotganini bildiradi.

srand(time(0)); - rand() funksiyasi har safar boshqa natija berishi uchun urug‘ (seed) sifatida hozirgi vaqtni beradi. Agar srand ishlatilmasa, rand() har safar bir xil sonlarni beradi.

for (int i = 0; i < sonlar_soni; ++i) {

int tasodifiy_son = rand() % 100;

cout << tasodifiy_son << " "; } - sonlar_soni marta aylanuvchi sikl. Har safar 0 dan 99 gacha tasodifiy butun son hosil qilinadi va ekranga chiqariladi.

cout << endl; - Yangi qatorga o‘tadi.

cout << endl << "2. Mersenne Twister (mt19937) yordamida:" << endl; - Bo‘lim 2: Zamonaviy Mersenne Twister algoritmiga asoslangan tasodifiy sonlar generatsiyasi boshlanmoqda.

mt19937 generator(static_cast<unsigned int>(time(0))); - mt19937 — yuqori sifatli psevdo-tasodifiy generator. Urug‘ sifatida vaqt (time(0)) berilmoqda.

uniform_int_distribution<int> butun_oraliq(0, 99); - dan 99 gacha bo‘lgan butun sonlar uchun bir xillikda taqsimlangan tasodifiy taqsimot.

for (int i = 0; i < sonlar_soni; ++i) {

int tasodifiy_son = butun_oraliq(generator);

cout << tasodifiy_son << " "; } - generator yordamida har safar yangi tasodifiy son hosil qilinadi va chiqariladi.

cout << endl << "3. Haqiqiy (real) sonlar [0.0, 1.0) oralig'ida:" << endl; - Bo‘lim 3: 0.0 dan 1.0 gacha bo‘lgan haqiqiy sonlar generatsiyasi boshlanganini bildiradi.

uniform_real_distribution<double> haqiqiy_oraliq(0.0, 1.0); - Bu haqiqiy sonlar uchun taqsimot obyektidir. Sonlar 0.0 (shu jumladan) dan 1.0 (chiqmaydigan) oralig‘ida bo‘ladi.

for (int i = 0; i < sonlar_soni; ++i) {

double haqiqiy_son = haqiqiy_oraliq(generator);

cout << haqiqiy_son << " "; } - Har bir aylanishda bitta haqiqiy tasodifiy son hosil qilinadi va chiqariladi.

return 0; - return 0 satri dasturning to'liq yakunlangani va xatolik yo'qligini bildiradi.

Dasturning ishlash prinsipi:

```
Nechta tasodifiy son generatsiya qilishni xohlaysiz? 5

1. Klassik rand() funksiyasi yordamida:
22 58 50 94 87

2. Mersenne Twister (mt19937) yordamida:
73 77 80 11 92

3. Haqiqiy (real) sonlar [0.0, 1.0) oralig'ida:
0.446523 0.655603 0.324331 0.698695 0.518286

Process returned 0 (0x0)   execution time : 1.943 s
Press any key to continue.
```

Ushbu dastur psevdo-tasodifiy sonlar generatsiyasini o‘z ichiga oladi, bu algoritmlar yordamida sonlar tizimli tarzda, lekin har safar yangilanib olinadi. Dasturning ishlash prinsipi tasodifiylikni ta'minlash orqali turli ilovalarda foydalanish uchun mos natijalar yaratishga imkon beradi. Algoritmlar orasida sifat va tezlik nuqtai nazaridan farqlar mavjud bo‘lib, zamonaviy metodlar ancha mukammal va ko‘p hollarda ishonchli hisoblanadi.

XULOSA

Ushbu dastur tasodifiy sonlar generatsiyasining turli usullarini ko‘rib chiqdi. Klassik rand() funksiyasi va zamonaviy **Mersenne Twister (mt19937)** algoritmi yordamida butun sonlar va haqiqiy (real) sonlar ishlab chiqarildi. Har ikkala metod

ham tasodifiy sonlar yaratish uchun keng tarqalgan usullar bo‘lib, ularning har biri o‘zining afzalliklari va cheklovlariga ega.

rand() funksiyasi sodda va ishlatish oson bo‘lsa-da, u ba'zan past sifatli natijalar bera oladi. Shu sababli, yuqori sifatli tasodifiy sonlar talab qilinadigan vazifalarda **Mersenne Twister** kabi ilg‘or algoritmlar afzalroq hisoblanadi. **Mersenne Twister** algoritmi yuqori tezlik va sifatni ta'minlab, uzoq muddat davomida xavfsiz va ishonchli natijalar yaratadi.

Shuningdek, haqiqiy (real) sonlar generatsiyasi ko‘proq tasodifiylik va aniqlikni talab qiladigan ilovalar uchun foydalidir, masalan, simulatsiyalar yoki modellashtirish.

Umuman olganda, dastur zamonaviy tasodifiy sonlar generatsiyasini tahlil qilib, uning amaliy ilovalarda qanday ishlashini ko‘rsatdi. Bu usullarni to‘g‘ri tanlash dastur sifatini va uning samaradorligini oshirishga yordam beradi.

FOYDALANILGAN ADABIYOTLAR:

1. C++ Programming Language - Bjarne Stroustrup
2. The Art of Computer Programming, Volume 2: Seminumerical Algorithms - Donald E. Knuth
3. C++ Standard Library - Nicolai M. Josuttis
4. Numerical Recipes: The Art of Scientific Computing - William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery