# *Національний технічний університет України*

***«Київський політехнічний інститут»***

#### ***Факультет інформатики та обчислювальної техніки***

### ***Кафедра обчислювальної техніки***

## ЛАБОРАТОРНА РОБОТА №1

*з курсу «Теорія ймовірності та математична статистика»*

***Виконала:*** *Косейкіна Ганна Сергіївна*

***Факультет***  *ІОТ*

***Група*** *ІО-91* **Залікова книжка****№** *9123*

***Задание***

Тип: Т(р12)

Разрядность: 6+

Объём выборки: 20000

Листинг:

**import** java.util.ArrayList;

public class Run {

**public** **static** **void** main(String[] args) {

**int** [] vyborka=**new** **int** [20000];

**int** [] pol1={1,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0};

**int** [] pol2={1,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0};

**int** [] pol3={1,0,1,1,1,0,0,0,1,0,0,0,0,0,0,0,0,0};

**int** [] pol4={1,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0};

**int** [] pol5={1,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0};

**int** [] pol6={1,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0};

**int** [] pol7={1,1,0,0,1,0,1,0,0,0,0,0,1,0,0,0,0,0};

**int** [] pol8={1,1,0,1,1,0,0,0,0,0,0,0,0,1,0,0,0,0};

**int** [] pol9={1,1,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0};

**int** [] pol10={1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0};

**int** [] pol11={1,1,0,1,0,0,0,0,0,0,0,0,1,0,0,0,1,0};

**int** [] pol12={1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1};

ArrayList <LFSR> lfsr=**new** ArrayList <LFSR>();

lfsr.add(0,**new** LFSR(pol1));

lfsr.add(1,**new** LFSR(pol2));

lfsr.add(2,**new** LFSR(pol3));

lfsr.add(3,**new** LFSR(pol4));

lfsr.add(4,**new** LFSR(pol5));

lfsr.add(5,**new** LFSR(pol6));

lfsr.add(6,**new** LFSR(pol7));

lfsr.add(7,**new** LFSR(pol8));

lfsr.add(8,**new** LFSR(pol9));

lfsr.add(9,**new** LFSR(pol10));

lfsr.add(10,**new** LFSR(pol11));

lfsr.add(11,**new** LFSR(pol12));

Generator generator=**new** Generator();

**int** s;

**for** (**int** i=0;i<vyborka.length;i++){

s=generator.generate(lfsr);

vyborka[i]=s;

System.*out*.print(s+".");

}

System.*out*.println("Tests: ");

System.*out*.println(GeneratorTester.*complexity*(vyborka));

System.*out*.println(GeneratorTester.*frequancyTest*(vyborka));

System.*out*.println(GeneratorTester.*xorFrequancyTest*(vyborka));

**int**[] ranks=GeneratorTester.*rankTest*(vyborka,3);

System.*out*.println("3) Rank test:");

**for** (**int** i=0; i<ranks.length; i++) {

System.*out*.println((**double**)ranks[i]/vyborka.length);

}

}

}

**import** java.util.Arrays;

public class GeneratorTester {

**public** **static** **double** frequancyTest(**int**[] arr){

**int** result=0;

**for**(**int** i=0; i<arr.length; i++){

**if**(arr[i]==1){

result++;

}

}

**return** ((**double**)result/arr.length);

}

**public** **static** **double** xorFrequancyTest(**int**[] arr){

**int** result=0;

**for**(**int** i=0; i<arr.length-1; i++){

**if**((arr[i]^arr[i+1])==1){

result++;

}

}

**return** ((**double**)result/(arr.length-1));

}

**public** **static** **int**[] rankTest(**int**[] arr, **int** width){

**if**(width<=0|width>10){

**throw** **new** IllegalArgumentException("Width must be range [2;10]");

}

**int**[] result=**new** **int**[(**int**)Math.*pow*(2,width)];

**int** window;

**for**(**int** i=0;i<arr.length-width;i++){

window=0;

**for** (**int** j=0; j<width; j++) {

window=window<<1;

window=window|arr[i+j];

}

result[window]++;

}

**return** result;

}

**public** **static** **void** FourthTest(**int**[] array) {

**final** **int** N = array.length;

**int**[] b = **new** **int**[N];

**int**[] c = **new** **int**[N];

**int**[] t = **new** **int**[N];

b[0] = 1;

c[0] = 1;

**int** l = 0;

**int** m = -1;

**for** (**int** n = 0; n < N; n++) {

**int** d = 0;

**for** (**int** i = 0; i <= l; i++) {

d ^= c[i] \* array[n - i];

}

**if** (d == 1) {

System.*arraycopy*(c, 0, t, 0, N);

**int** N\_M = n - m;

**for** (**int** j = 0; j < N - N\_M; j++) {

c[N\_M + j] ^= b[j];

}

**if** (l <= n / 2) {

l = n + 1 - l;

m = n;

System.*arraycopy*(t, 0, b, 0, N);

}

}

}

System.*out*.println("Fourth Test result:" + l);

}

}

**import** java.util.ArrayList;

**import** java.util.Random;

public class Generator {

**private** **int** [] tabl=**new** **int**[4096];

**private** **int** pos=0;

**public** Generator(){

**for** (**int** i=0;i<tabl.length;i++){

Random r=**new** Random();

tabl[i]=r.nextInt(2);

// System.out.print(tabl[i]);

}

}

**public** **int** generate(ArrayList <LFSR> lfsr){

**int** res=0;

**this**.pos=0;

**int** []mas=**new** **int**[12];

**for**(**int** i=0;i<mas.length;i++){

mas[i]=lfsr.get(i).generate();

}

**for** (**int** i=0;i<mas.length;i++){

**if** (mas[i]==1){

pos=pos+(**int**)(Math.*pow*(2, i));

}

}

res=tabl[pos];

//System.out.println(res);

**return** res;

}

}

**public class LFSR** {

**int** []razr={1,1,1,0,0,0,1,0,1,1,1,1,0,0,0,0,1,0};

**int** gen;

**int** [] pol;

**public** **static** **int** xor(**int** a,**int** b){

**if**((a+b)==2){

**return** 0;

}

**if**((a+b)==1){

**return** 1;

}

**return** 0;

}

**public** LFSR(**int** [] pol){

**int** res=**this**.razr[17];

**for**(**int** i=17;i>0;i--){

**if**(pol[i-1]==1){

**this**.razr[i]=*xor*((**this**.razr[i-1]),res);

}**else**{

**this**.razr[i]=**this**.razr[i-1];

}

}

**this**.razr[0]=res;

**this**.pol=pol;

}

**public** **int** generate(){

**int** res=**this**.razr[17];

**for**(**int** i=17;i>0;i--){

**if**(**this**.pol[i-1]==1){

**this**.razr[i]=*xor*((**this**.razr[i-1]),res);

}**else**{

**this**.razr[i]=**this**.razr[i-1];

}

}

**this**.razr[0]=res;

gen=res;

**return** gen;

}

}