



Міністерство освіти і науки України

Національний технічний університет України «КПІ

імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

ЗВІТ

лабораторної роботи №1

з дисципліни «Моделювання систем»

Перевірила:

Дифучина О. Ю.

Виконав:

Студент Гр. ІІІ-01

Пашковський Є. С.

Київ 2023

Завдання

- ✓ Згенерувати 10000 випадкових чисел трьома вказаними нижче способами. **45 балів.**

- Згенерувати випадкове число за формулою $x_i = -\frac{1}{\lambda} \ln \xi_i$, де ξ_i - випадкове число, рівномірно розподілене в інтервалі (0;1). Числа ξ_i можна створювати за допомогою вбудованого в мову програмування генератора випадкових чисел. Перевірити на відповідність експоненційному закону розподілу $F(x) = 1 - e^{-\lambda x}$. Перевірку зробити при різних значеннях λ .
- Згенерувати випадкове число за формулами:

$$x_i = \sigma \mu_i + a$$

$$\mu_i = \sum_{j=1}^{12} \xi_j - 6,$$

де ξ_i - випадкове число, рівномірно розподілене в інтервалі (0;1). Числа ξ_i можна створювати за допомогою вбудованого в мову програмування генератора випадкових чисел. Перевірити на відповідність нормальному закону розподілу:

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(x-a)^2}{2\sigma^2}\right).$$

Перевірку зробити при різних значеннях a і σ .

- Згенерувати випадкове число за формулою $z_{i+1} = az_i \pmod{c}$, $x_{i+1} = z_{i+1}/c$, де $a=5^{13}$, $c=2^{31}$. Перевірити на відповідність рівномірному закону розподілу в інтервалі (0;1). Перевірку зробити при різних значеннях параметрів a і c .
- ✓ Для кожного побудованого генератора випадкових чисел побудувати гістограму частот, знайти середнє і дисперсію цих випадкових чисел. По виду гістограми частот визначити вид закону розподілу. **20 балів.**
- ✓ Відповідність заданому закону розподілу перевірити за допомогою критерію згоди χ^2 . **30 балів**
- ✓ Зробити висновки щодо запропонованих способів генерування випадкових величин. **5 балів**

Хід роботи

Код серверу:

```
import express, { Request, Response } from 'express';
import cors from 'cors';
import generateRandomOne from './generateRandomOne';
import generateRandomTwo from './generateRandomTwo';
import generateRandomThree from './generateRandomThree';
import exponentialSegment from './exponentialSegment';

const k = 100;

const app = express();

app.use(cors());

const testGenerator =
  (generate: () => number, testFunc: (x1: number, x2: number) => number) =>
  (req: Request, res: Response) => {
    const numbers: number[] = [];

    const n = 10000;

    for (let i = 0; i < n; i++) {
      numbers.push(generate());
    }

    const max = Math.max(...numbers);
    const min = Math.min(...numbers);

    const h = (max - min) / k;

    const countTable: { [key: number]: number } = {};

    const i0 = min;

    for (let m = 0; m < k; m++) {
      const i = i0 + m * h;

      const index = i + h / 2;
      countTable[index] = numbers.filter(
        (num) => num >= i && num < i + h
      ).length;
    }

    const freqTable = Object.entries(countTable)
      .map((item) => ({
        x: +item[0],
        y: item[1],
      })))
```

```

        .sort((a, b) => a.x - b.x);

let X2 = 0;
let X2k = 0;
let M = 0;
let M2 = 0;

console.table(
  freqTable.map(({ x, y }) => {
    const yExpected = testFunc(x - h / 2, x + h / 2) * n;
    const X2part =
      y >= 5 ? ((y - yExpected) * (y - yExpected)) / yExpected : 0;

    X2 += X2part;
    if (y >= 5) X2k++;

    const freq = y / n;

    M += x * freq;
    M2 += x * x * freq;

    return { x, yGot: y, yExpected, diff: Math.abs(yExpected - y), X2part };
  })
);

console.log(`M=${M}\nD=${M2 - M * M}\nX2 (k=${X2k}) = ${X2}`);

res.json(freqTable);
};

app.get(
  '/one',
  testGenerator(() => generateRandomOne(1), exponentialSegment(1))
);

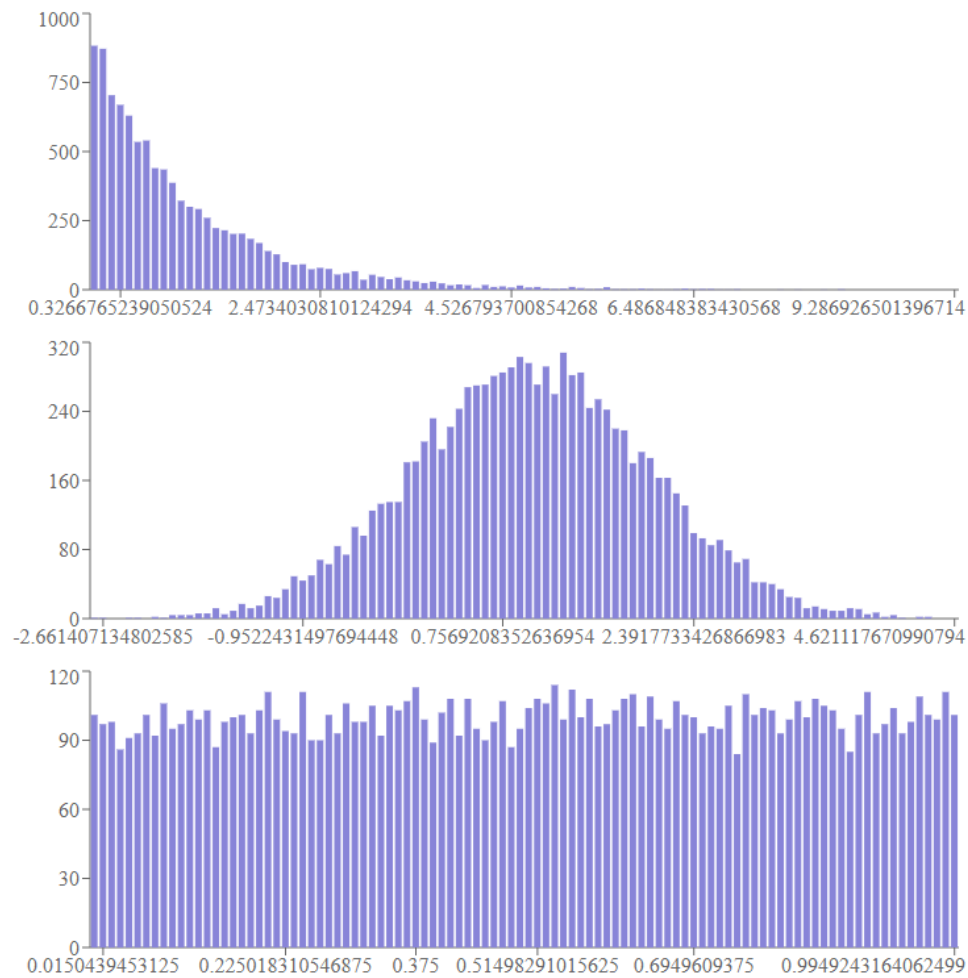
app.get(
  '/two',
  testGenerator(
    () => generateRandomTwo(1, 1),
    () => 0
  )
);

app.get(
  '/three',
  testGenerator(
    () => generateRandomThree(),
    () => 1 / k
  )
);

```

```
app.listen(5000, () => {  
  console.log('server started');  
});
```

Результати роботи:



83	0.83494384765625	108	100	8	0.64
84	0.8449426269531249	105	100	5	0.25
85	0.8549414062499999	103	100	3	0.09
86	0.8649401855468749	95	100	5	0.25
87	0.87493896484375	85	100	15	2.25
88	0.884937744140625	101	100	1	0.01
89	0.8949365234375	111	100	11	1.21
90	0.904935302734375	93	100	7	0.49
91	0.9149340820312499	97	100	3	0.09
92	0.9249328613281249	104	100	4	0.16
93	0.934931640625	93	100	7	0.49
94	0.944930419921875	98	100	2	0.04
95	0.95492919921875	109	100	9	0.81
96	0.964927978515625	101	100	1	0.01
97	0.9749267578124999	99	100	1	0.01
98	0.9849255371093749	111	100	11	1.21
99	0.9949243164062499	101	100	1	0.01

M=0.5034228168945313
D=0.08260414810599881
X2 (k=100) = 46.69000000000001

Висновки: під час виконання цього завдання було досліджено різні способи генерування випадкових чисел та протестовано їх роботу.