Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра Математической кибернетики

и компьютерных наук

**ЛАБОРАТОРНАЯ РАБОТА №6**

По дисциплине «Моделирование»

студента 4 курса 411 группы

направления ФИИТ

факультета компьютерных наук и информационных технологий

Гайворонского Алексея

Саратов 2021

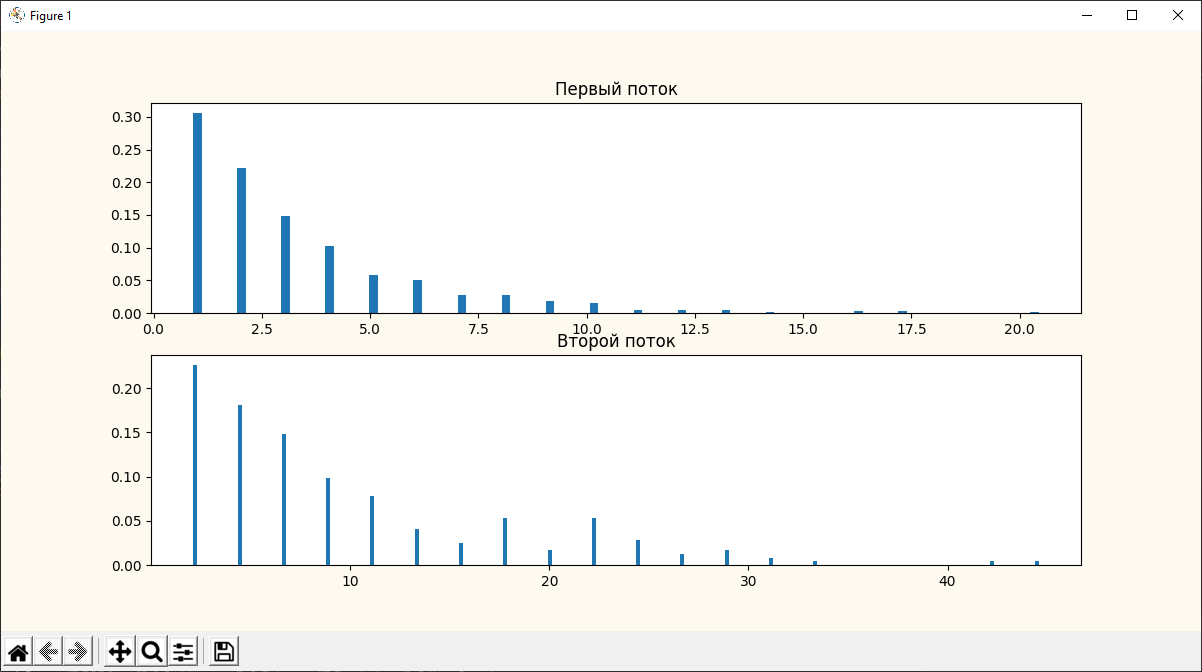
**Задача 3**. Пуассоновский поток частиц разделяется на два потока таким образом, что в первый поток частицы попадают с вероятностью 3/4, а во второй с вероятностью 1/4. Построить модель потока и разделения частиц. Получить выборку из 1000 моментов поступления частиц. Построить выборочную функцию плотности распределения длительности интервала времени между двумя частицами для каждого из потоков.

**Код:**

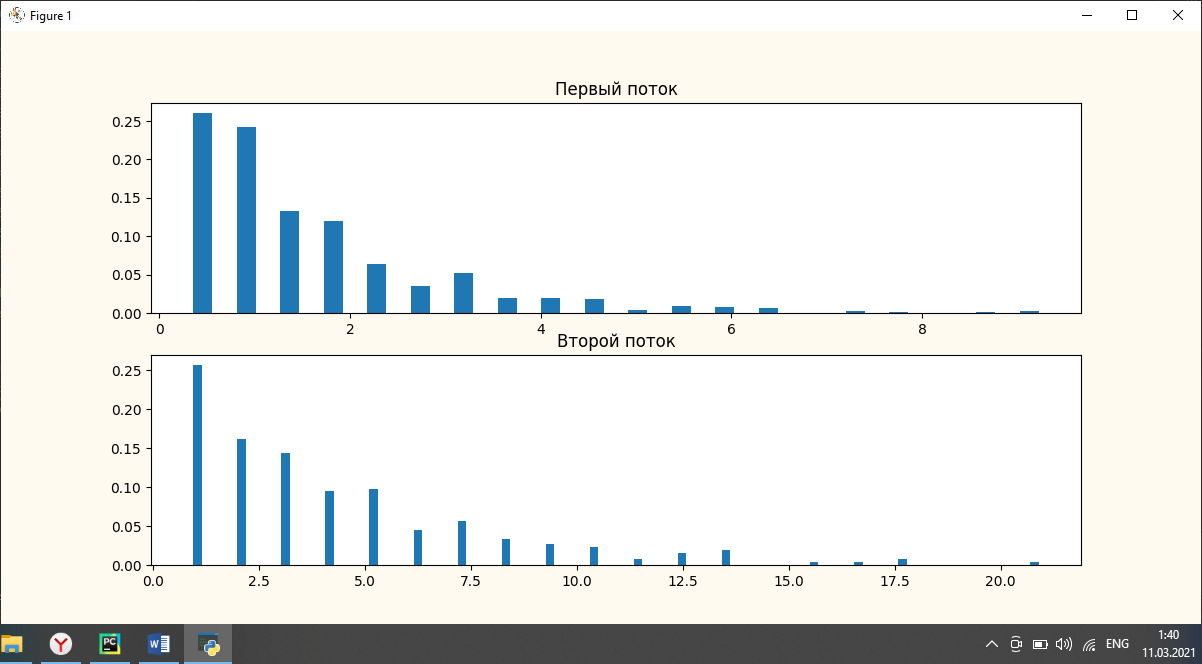
import random  
import math  
import matplotlib.pyplot as plt  
from numpy.core.function\_base import linspace  
  
N = 1000 # число частиц  
LMBD = 2 # лямбда  
  
  
def puasson():  
 return -1 / LMBD \* math.log(random.random())  
  
  
def get\_intervals(temp\_list: list) -> list:  
 return [abs(temp\_list[\_i + 1] - temp\_list[\_i]) for \_i in range(len(temp\_list) - 1)]  
  
  
def calc\_intervals\_range(stream: list) -> dict:  
 max\_x = max(stream)  
 intervals = {}  
 for interval\_end in linspace(0, max\_x, 21)[1:]: # нулевой интервал следует выбросить  
 intervals[interval\_end] = []  
  
 for x in stream:  
 for interval in intervals:  
 if x <= interval:  
 intervals[interval].append(x)  
 break  
  
 return intervals  
  
  
def calc\_frequencies(intervals: dict, count\_intervals: int) -> dict:  
 frequencies = {}  
 for interval, dots in intervals.items():  
 frequencies[interval] = len(dots) / count\_intervals  
  
 return frequencies  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 t = 0  
 first = []  
 second = []  
 for i in range(N):  
 if random.random() <= 0.75:  
 first.append(t) # 3/4 частиц - в первый поток  
 else:  
 second.append(t)  
 t += puasson()  
 xs = get\_intervals(first)  
 ys = get\_intervals(second)  
 xs\_len = len(xs)  
 ys\_len = len(ys)  
  
 first\_intervals = calc\_intervals\_range(xs)  
 second\_intervals = calc\_intervals\_range(ys)  
  
 first\_freqs = calc\_frequencies(first\_intervals, xs\_len)  
 second\_freqs = calc\_frequencies(second\_intervals, ys\_len)  
  
 fig, axes = plt.subplots(2, 1)  
  
 axes[0].bar(first\_freqs.keys(), first\_freqs.values(), width=0.1)  
 axes[1].bar(second\_freqs.keys(), second\_freqs.values(), width=0.3)  
  
 axes[0].set\_title('Первый поток')  
 axes[1].set\_title('Второй поток')  
  
 fig.set\_facecolor('floralwhite')  
 fig.set\_figwidth(12) # ширина Figure  
 fig.set\_figheight(6) # высота Figure  
  
 plt.show()

**Результаты:**

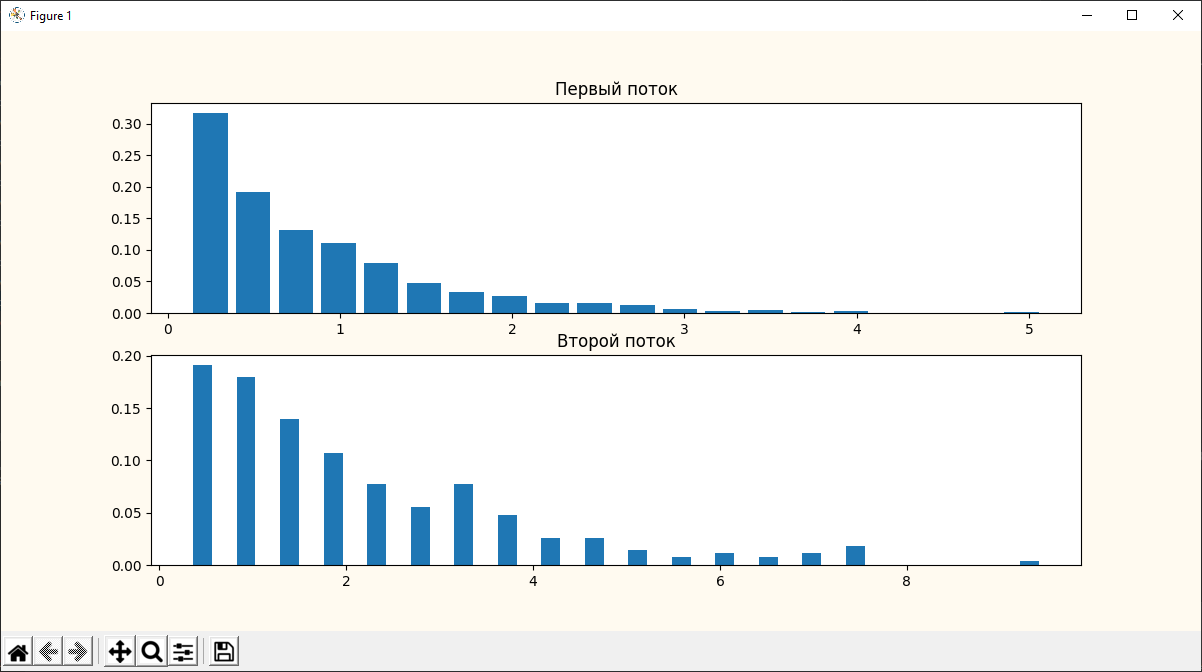
LMBD = 0.5



LMBD = 1



LMBD = 2



LMBD = 3

