Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

Дисциплина «Высокопроизводительные вычисления»

**Лабораторная работа № 1**

**Тестирование псевдослучайных последовательностей**

Выполнил:

Студент группы ИВТАСбд-42

Сулейманов М.З.

Проверил:

Мартынов А.И.

Ульяновск

2024

**Цель работы**

Научиться тестировать последовательность бит на равномерность и случайность. А именно: сделать программу на любом языке программирования выполняющую частотный тест.

Задание

Реализовать приложение, позволяющее выполнять следующие действия:

1. Задавать длину генерируемой последовательность в битах (при тестировании рекомендуется задавать длину последовательности не менее 10 000 бит)

2. Генерировать псевдослучайную последовательность 0 и 1 с помощью стандартного алгоритма генерации случайных чисел

3. Загружать последовательность из текстового файла

4. Сохранять полученную последовательность в файл и выводить ее на экран приложения

5. Проверять полученную последовательность с помощью реализованных тестов. Результат проверки должен отображаться в приложении

**Частотный тест**

Этот тест оценивает пропорцию нулей и единиц в проверяемой последовательности. Тест определяет, является ли количество нулей и единиц в последовательности приблизительно таким же, как должно быть в истинно случайной последовательности.

Шаги алгоритма:

1. Входная последовательность, состоящая из 0 и 1 (будем обозначать ее ε), преобразовывается в последовательность -1 и 1 (будем обозначать ее X) соответственно:

𝑋𝑖 = 2 ∗ 𝜀𝑖 − 1

2. Вычисляется сумма 𝑆𝑛 = 𝑋1 + 𝑋2 + 𝑋3 + ⋯ + 𝑋𝑛, где n – количество элементов проверяемой последовательности.

3. Вычисляется статистика 𝑆 =|𝑆𝑛|/√𝑛

4. Если 𝑆 ≤ 1.82138636, то тест считается успешно пройденным, иначе делается вывод о том, что последовательность является неслучайной.

**Листинг**

|  |
| --- |
| from tkinter import \*  from tkinter import Tk, RIGHT, BOTH, RAISED  from tkinter.ttk import Frame, Button, Style  from textwrap import wrap  from tkinter import ttk  import random  import math  global s #глобальная переменная которая будет содержать последовательность символов  s="1" #Просто чтобы пустой не была вдруг поломается что-то  sizeof=0  def generate\_random\_bits(num\_bits): #Функция для генерации последовательности      bits = []      for \_ in range(num\_bits):          bit = random.randint(0, 1)          bits.append(str(bit))      bit\_string = ''.join(bits)      return bit\_string  def generate(): #Функция для кнопки генерации последовательности      global s #Вызов глобальной перменной      sizeof=int(sizeEntry.get()) #получение размера последовательности из поля для ввода      if(sizeof!=0):          s = generate\_random\_bits(sizeof) #присвоение значения глобальной переменной          lbl.delete("1.0", END) # Очищаем Text виджет          lbl.insert(END, s) # Вставляем текст в Text виджет          with open("file.txt", "w") as output: #Открытие файла              output.write(s) #Запись      else:          lbl.delete("1.0", END) # Очищаем Text виджет          lbl.insert(END, "Введите значение в поле справа от кнопки Сгенерировать") # Вставляем текст в Text виджет  def load(): #Функция для кнопки загрузки последовательности из файла      global s #Вызов глобальной перменной      lbl.delete("1.0", END) # Очищаем Text виджет      lbl.insert(END, "Загружено") # Вставляем текст в Text виджет      with open("file.txt", "r") as input: #Открытие файла          s=input.read() #Чтение      lbl.delete("1.0", END) # Очищаем Text виджет      lbl.insert(END, s) # Вставляем текст в Text виджет  def check(): #Функция для кнопки проверки загруженной/сгенерированной последовательности      global s #Вызов глобальной перменной      sum=0 #Будущая сумма чисел последовательности      for i in range(len(s)): #Получение суммы чисел последовательности          if(s[i])=='1':              sum+=1          else:              sum-=1      if((abs(sum)/math.sqrt(len(s)))<1.82138636): #Проверка последовательности          lbl.delete("1.0", END) # Очищаем Text виджет          lbl.insert(END, "Частотный тест пройден успешно!") # Вставляем текст в Text виджет      else:          lbl.delete("1.0", END) # Очищаем Text виджет          lbl.insert(END, "Последовательность не прошла частотный тест") # Вставляем текст в Text виджет  #Создание окна и кнопок  window = Tk()  window.title("Частотный тест")  window.geometry('800x620')  lbl = Text(window, font=("Arial", 10), wrap=WORD) # wrap=WORD для переноса слов  lbl.pack(fill=BOTH, expand=True)  scrollbar = ttk.Scrollbar(window, orient="vertical", command=lbl.yview)  scrollbar.pack(side=RIGHT, fill=Y)  lbl.config(yscrollcommand=scrollbar.set) # Связываем Text с полосой прокрутки  sizeEntry=Entry(font=("Arial",10), text="asdasd")  sizeEntry.pack(side=RIGHT)  generateButton = Button(text="Сгенерировать", command=generate)  generateButton.pack(side=RIGHT, padx=5, pady=5)  loadButton = Button(text="Загрузить", command=load)  loadButton.pack(side=RIGHT)  generateButton = Button(text="Проверить", command=check)  generateButton.pack(side=RIGHT, padx=5, pady=5)  window.mainloop() |