Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

Кафедра вычислительных систем

**Лабораторная работа №2**

«Выборка без возвращения»

Выполнил:

студент гр. МГ-211 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Бурдуковский И.А./

подпись

Проверил:

Профессор

кафедры ВС \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Родионов А.С./

Новосибирск

2023 г.

Оглавление

[Задание 3](#_Toc133933349)

[Описание работы 4](#_Toc133933350)

# Задание

Написать программу для выборки без возвращения

# Описание работы

В соответствии с заданием была создана python программа. При запуске она генерирует набор вероятностей, на основе которого создаётся индексный массив. Далее, в зависимости от указанного количества прогонов, срабатывает алгоритм:

1. Индексный массив перемешивается случайным образом
2. Из него последовательно отбираются уникальные индексы
3. Данные значения записываются в итоговую выборку

Для анализа результата выборок и вывода статистики была использована балльная система: элемент получает количество баллов, равное разнице количества выбираемых элементов и номера позиции элемента в выборке. Полученные за одну выборку баллы суммируются в итоговый счёт вероятности.

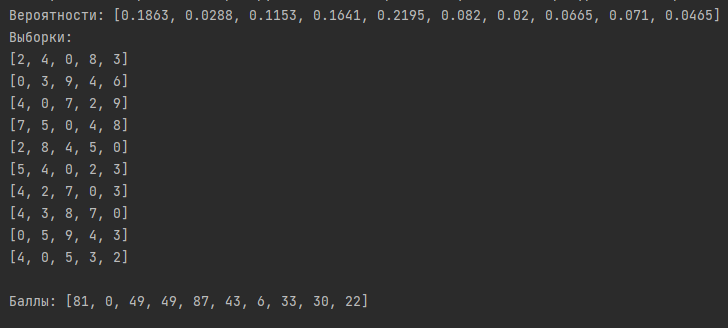


Рис. 1 - Выборка 5 элементов из 10.   
Количество выборок: 10

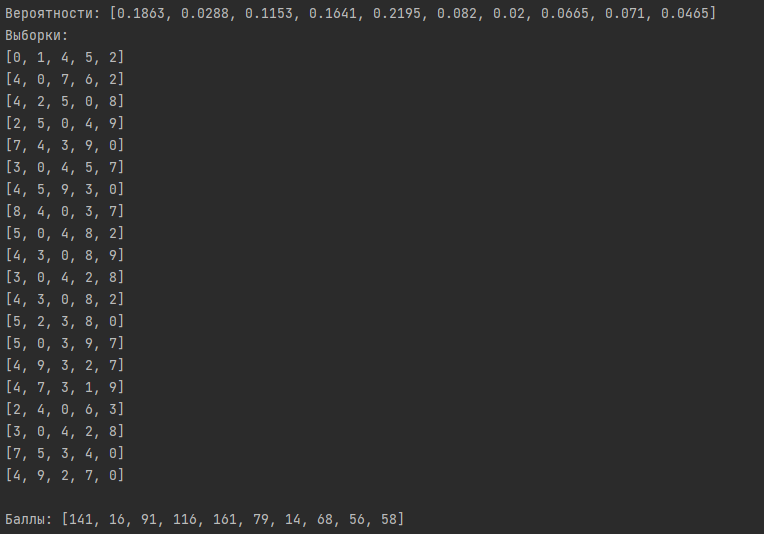


Рис. 2 - Выборка 5 элементов из 10.   
Количество выборок: 20

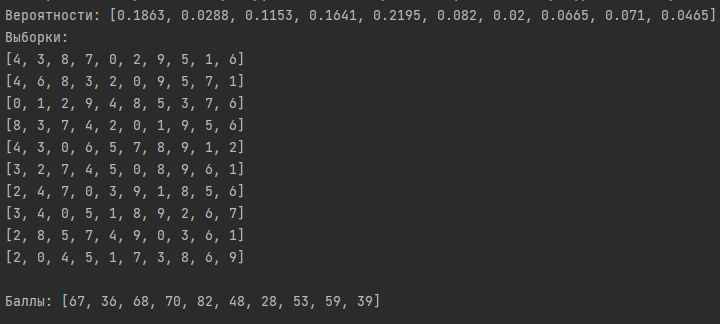


Рис. 3 - Выборка 10 элементов из 10.   
Количество выборок: 10

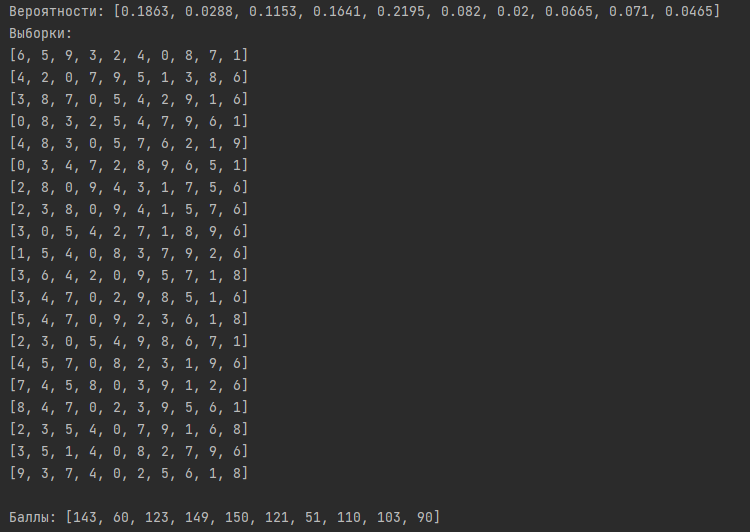


Рис. 4 - Выборка 10 элементов из 10.   
Количество выборок: 20

На результатах работы программы с разными параметрами можно пронаблюдать, что наибольшей вероятности соответствует наибольше количество баллов в итоговом счете. И наоборот – наименьшие вероятности почти не набрали баллов. Однако, с увеличением количества выбираемых элементов (вплоть до всех) – уменьшается разница по набранным баллам между разными вероятностями.

# Листинг

**import random  
  
def generate\_random\_propability(N: int):  
 propabilities = []  
 sum = 0  
 check\_sum = 0  
  
 for i in range(N):  
 propabilities.append(random.randint(1, 100))  
 sum = sum + propabilities[i]  
  
 for i in range(N):  
 check\_sum = check\_sum + round(propabilities[i] / sum, 4)  
 propabilities[i] = round(propabilities[i] / sum, 4)  
  
 if check\_sum != 1.:  
 propabilities[-1] = round(propabilities[-1] + 1. - check\_sum, 4)  
  
 return propabilities  
  
  
def select\_without\_repeats(propabilities\_list: list, selection\_size):  
 index\_counters = []  
 index\_mass = []  
 for i in propabilities\_list:  
 index\_counters.append(int(i \* 10000))**

**for i in range(len(index\_counters)):  
 for j in range(index\_counters[i]):  
 index\_mass.append(i)**

**random.shuffle(index\_mass)  
 selection = []  
 for i in range(selection\_size):  
 selection.append(index\_mass[0])  
 new\_index\_mass = []  
 for j in range(len(index\_mass)):  
 if index\_mass[j] != selection[-1]:  
 new\_index\_mass.append(index\_mass[j])  
 index\_mass = new\_index\_mass  
  
 return selection  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 propabilities = generate\_random\_propability(10)**

**print("Вероятности:", propabilities)  
 runs\_count = 20  
 rating = []  
 for i in range(len(propabilities)):  
 rating.append(0)  
  
 print("Выборки:")  
 for i in range(runs\_count):  
 selection = select\_without\_repeats(propabilities, 10)  
 for j in range(len(selection)):  
 rating[selection[j]] = rating[selection[j]] + (len(propabilities) - j)  
 print(selection)  
  
 print("\nБаллы:", rating)**