

Санкт-Петербургский Национальный Исследовательский Университет
Информационных технологий, механики и оптики

Факультет инфокоммуникационных технологий

Домашняя работа №5

Выполнили:

Смирнов И.И.

Телунц Э.Р.

Царёв А.С.

Проверил:

Мусаев А.А.

Санкт-Петербург

2023

СОДЕРЖАНИЕ

Стр.

ВВЕДЕНИЕ	3
1 Задание 1	4
2 Задание 2	5
3 Задание 3	6
4 Задание 4	7
5 Задание 5	8
6 Задание 6	9
7 Задание 7	10
ЗАКЛЮЧЕНИЕ	11
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	12

ВВЕДЕНИЕ

Для становления хорошим специалистом в области программирования на языке Python необходимо знать основные алгоритмы и функционал языка.

Цель данной работы – реализация различных алгоритмов на языке Python.

В ходе лабораторной работы были решены следующие задачи:

- написание кода для решения поставленных заданий;

Задания, которые необходимо выполнить:

1) Разработать алгоритм, проверяющий результат игры в крестики-нолики (3x3);

2) Для заданной матрицы $M \times N$, в которой каждая строка и столбец отсортированы по возрастанию, написать метод поиска элемента;

3) Написать алгоритм, находящий все варианты расстановки восьми ферзей на шахматной доске размером 8x8 так, чтобы никакие две фигуры не располагались на одной горизонтали, вертикали или диагонали (учитываются не только две главные, но и все остальные диагонали);

4) Ребенок поднимается по лестнице из n ступенек. За один шаг он может переместиться на одну, две или три ступеньки. Реализовать метод, рассчитывающий количество возможных вариантов перемещения ребенка по лестнице;

5) Описать, как бы вы использовали один одномерный массив для реализации трех стеков;

6) Написать максимально короткий код для экспоненциального фильтра;

7) Дан неотсортированный массив целых чисел. Вернуть наименьшее пропущенное целое число. Алгоритм должен выполняться за время $O(n)$

1 Задание 1

В задании необходимо написать алгоритм, проверяющий результат игры в крестики-нолики. Для этого на ввод подается результат игры (поле, заполненное крестиками и ноликами), после чего выводится итог партии.

```
Введите 1 строку: x o x
Введите 2 строку: o o x
Введите 3 строку: x o o
Победили нолики!
```

Рисунок 1 - Вывод задания 1

2 Задание 2

В этом задании нужно найти элемент в матрице $M \times N$, строки и столбцы которой отсортированы по возрастанию. Для этого подается отсортированная матрица, после чего запускается алгоритм поиска, который находит расположение числа в матрице (строка, столбец) или утверждает, что числа в матрице нет.

```
Введите кол-во строк: 2
Введите кол-во столбцов: 3
Введите элемент: 1
Введите элемент: 2
Введите элемент: 3
Введите элемент: 6
Введите элемент: 7
Введите элемент: 8
Введите число, которое нужно найти: 6
(2, 1)
```

Рисунок 2 - Вывод задания 2

3 Задание 3

В данном задании необходимо найти все варианты расстановки 8 ферзей на шахматной доске 8x8 так, чтобы ни одна пара ферзей не стояла на одной горизонтали, вертикали или диагонали. Для этого создается "пустое" поле, далее происходят расчеты, в ходе которых выясняется можно ли поставит фигуру так, чтобы условие выполнялось. В случае успешной постановки вместо 0 ставится цифра 2, а когда на доске появляется 8 ферзей, счетчик прибавляется и процесс начинается заново.

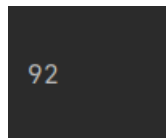
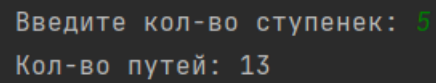


Рисунок 3 - Вывод задания 3

4 Задание 4

В задании необходимо просчитать все возможные варианты подняться в n количеству лестниц, при условии, что можно за шаг переходить 1, 2 или 3 ступеньки. Для этого написана функция на основе рекурсии.



```
Введите кол-во ступенек: 5
Кол-во путей: 13
```

Рисунок 4 - Вывод задания 4

5 Задание 5

В задании необходимо описать использование одномерного массива для реализации трех стеков. Чтобы это сделать нужно разделить 1 одномерный массив на 3 части, каждая из которых будет являться стеком.

```
nums = [] #одномерный список
class ThreeStack:
    def __init__(self):
        self.nums = []
        self.border1 = 0
        self.border2 = 0

    def insert(self, num_of_stack, num):
        if (num_of_stack == 1):
            self.border1 += 1
            self.border2 += 1
            self.nums.insert(self.border1 - 1, num)
        elif (num_of_stack == 2):
            self.border2 += 1
            self.nums.insert(self.border2 - 1, num)
        elif (num_of_stack == 3):
            nums.append(num)

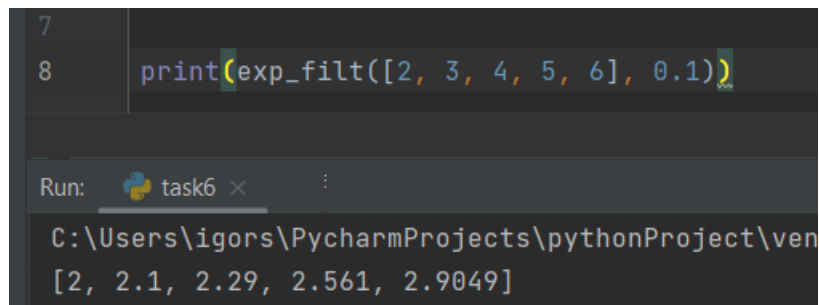
    def printinfo(self):
        print(self.nums)

stack = ThreeStack()
stack.insert(1, 1)
stack.insert(1, 2)
stack.insert(1, 3)
stack.insert(2, 6)
stack.insert(2, 9)
stack.insert(3, 1)
stack.insert(1, 7)
stack.printinfo()
```

Рисунок 5 - код задания 5

6 Задание 6

В задании необходимо написать максимально короткий код для экспоненциального фильтра. Была написана функция, которая проходит по массиву и с помощью специальной формулы для экспоненциального сглаживания возвращает новые значения.



```
7  
8 print(exp_filt([2, 3, 4, 5, 6], 0.1))
```

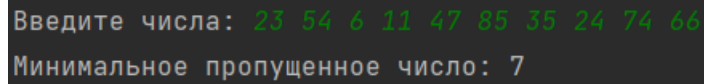
Run: task6 ×

C:\Users\igors\PycharmProjects\pythonProject\ven
[2, 2.1, 2.29, 2.561, 2.9049]

Рисунок 6 - Вывод задания 6

7 Задание 7

В задании необходимо вернуть наименьшее пропущенное число в неотсортированном массиве целых чисел. Программа должна работать за время $O(n)$. Программа принимает последовательность чисел, выбирает среди них максимальное и минимальное и начинает с помощью цикла проходить по всем числам от минимального до максимального. Первое число, которое не окажется в множестве будет являться наименьшим пропущенным числом.



```
Введите числа: 23 54 6 11 47 85 35 24 74 66  
Минимальное пропущенное число: 7
```

Рисунок 7 - Вывод задания 7

ЗАКЛЮЧЕНИЕ

В данной лабораторной работе был получен опыт работы на языке Python, а также были закреплены навыки использования алгоритмов и структур данных на языке Python .

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Wikipedia: официальный сайт: https://translated.turbopages.org/proxy_u/en-ru.ru.c0d61a88-6475c3cf-c1f9a14a-74722d776562/https/en.wikipedia.org/wiki/Exponential_smoothing (Дата обращения 30.05.2023)

Ссылка на полный код

<https://github.com/Blaaaaaaze/Algoritms/tree/main/lab10>