

Санкт-Петербургский Национальный Исследовательский
Университет Информационных Технологий, Механики и Оптики

Факультет инфокоммуникационных технологий

Лабораторная работа №8 "Асинхронное программирование"

Выполнили:
Голованов Д.И.,
Шарыпов Е.А.,
Стафеев И.А.
Проверил
Мусаев А.А.

Санкт-Петербург,
2024

СОДЕРЖАНИЕ

Стр.

ВВЕДЕНИЕ	3
1 Задача 1	4
2 Задача 2	5
3 Задача 3	6
4 Задачи с литкода	7
4.1 Задача Егора	7
4.2 Задача Димы	9
4.3 Задача Ивана	11
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	13

ВВЕДЕНИЕ

В данной лабораторной работе необходимо решить следующие задачи:

1. Написать программу, которая параллельно загружает несколько веб-страниц с использованием библиотеки `asynсio`. Программа должна быть способна обрабатывать несколько запросов одновременно и выводить результаты загрузки.
2. Создайте асинхронного клиента для работы с API. Реализуйте логику запроса данных к API, их обработки и вывода результатов, используя `async/await` синтаксис в Python.
3. Разработать программу, которая проводит одновременную обработку и анализ нескольких файлов с большим объемом данных. Использовать библиотеку `asynсio` для создания асинхронных задач по чтению и обработке файлов, чтобы повысить эффективность программы.

1 Задача 1

Был написан скрипт для асинхронных запросов страниц с википедии [1] с помощью библиотеки requests [2]. Ниже приведён его исходный код 1.1.

```
import requests
import asyncio

articles = ['Banana', 'Apple', 'Orange_(fruit)']

async def wiki_req(article):
    return requests.get(f'https://en.wikipedia.org/wiki/{article}').text

async def wiki_req_many(articles):
    return await asyncio.gather(*[wiki_req(article) for article in articles])

async def main():
    pages = await wiki_req_many(articles)
    for page in pages:
        print(page[:100])
        print()

asyncio.run(main())
```

Рисунок 1.1 — Скрипт для асинхронных запросов к Википедии

2 Задача 2

Нами был создан бот ИТМО.Савелий [7] в Телегаме, который помогает готовиться к экзамену по истории. Для генерации ответов был использован API YandexGPT [6], запущенный на площадке Yandex.Cloud [5]. Для взаимодействия с API Telegram была использована библиотека python-telegram-bot [4]. Полный код можно посмотреть в гитхаб-репозитории [8].

3 Задача 3

Создана программа, асинхронно открывающая файлы и заменяющая подстроку a1 на подстроку a2. Ниже приведён его исходный код 3.1.

```
.py / ...
import asyncio

async def replace_substring(filename, a1, a2):
    with open(filename, "r") as file:
        text = file.read()
    with open(filename, "w") as file:
        file.write(text.replace(a1, a2))

async def main():
    task1_handle = asyncio.ensure_future(replace_substring("a.txt", "a", "b"))
    task2_handle = asyncio.ensure_future(replace_substring("b.txt", "b", "a"))

    await asyncio.gather(task1_handle, task2_handle)

asyncio.run(main())
```

Рисунок 3.1 — Описание задачи

4 Задачи с литкода

4.1 Задача Егора

Была решена задача 84 [9]. Ниже приведены её описание 4.1, решение 4.2 и результат 4.3.

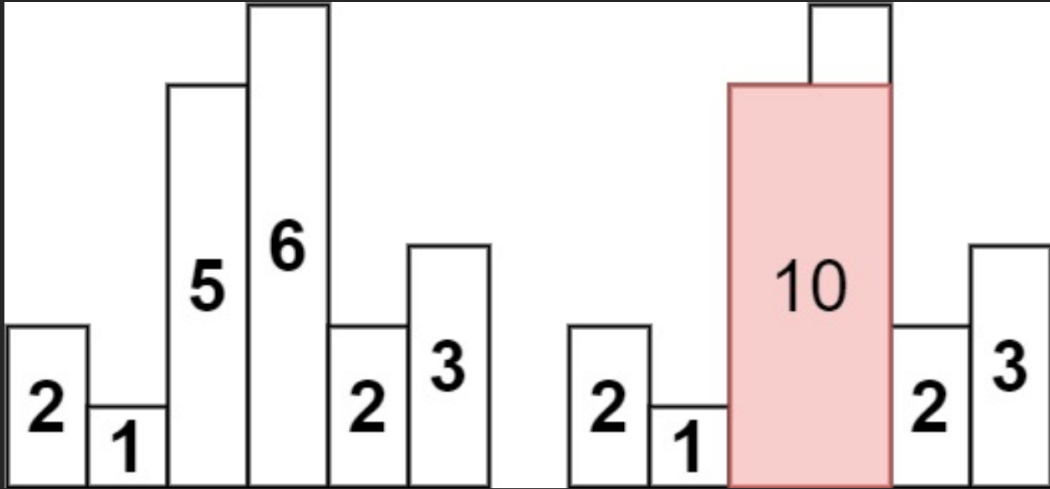
84. Largest Rectangle in Histogram

Solved ✓

Hard Topics Companies

Given an array of integers `heights` representing the histogram's bar height where the width of each bar is `1`, return *the area of the largest rectangle in the histogram*.

Example 1:



Input: `heights = [2,1,5,6,2,3]`
Output: `10`
Explanation: The above is a histogram where width of each bar is 1. The largest rectangle is shown in the red area, which has an area = 10 units.

Рисунок 4.1 — Описание задачи

```

class Solution:
    def largestRectangleArea(self, heights: List[int]) -> int:
        stack = [[0, 0]]
        result = 0
        heights.append(0)
        for i, h in enumerate(heights):
            s = i
            while stack[-1][0] > h:
                h1, i1 = stack.pop()
                result = max(result, h1 * (i - i1))
                s = i1
            if h > stack[-1][0]:
                stack.append((h, s))
        return result

```

Рисунок 4.2 — Код решения

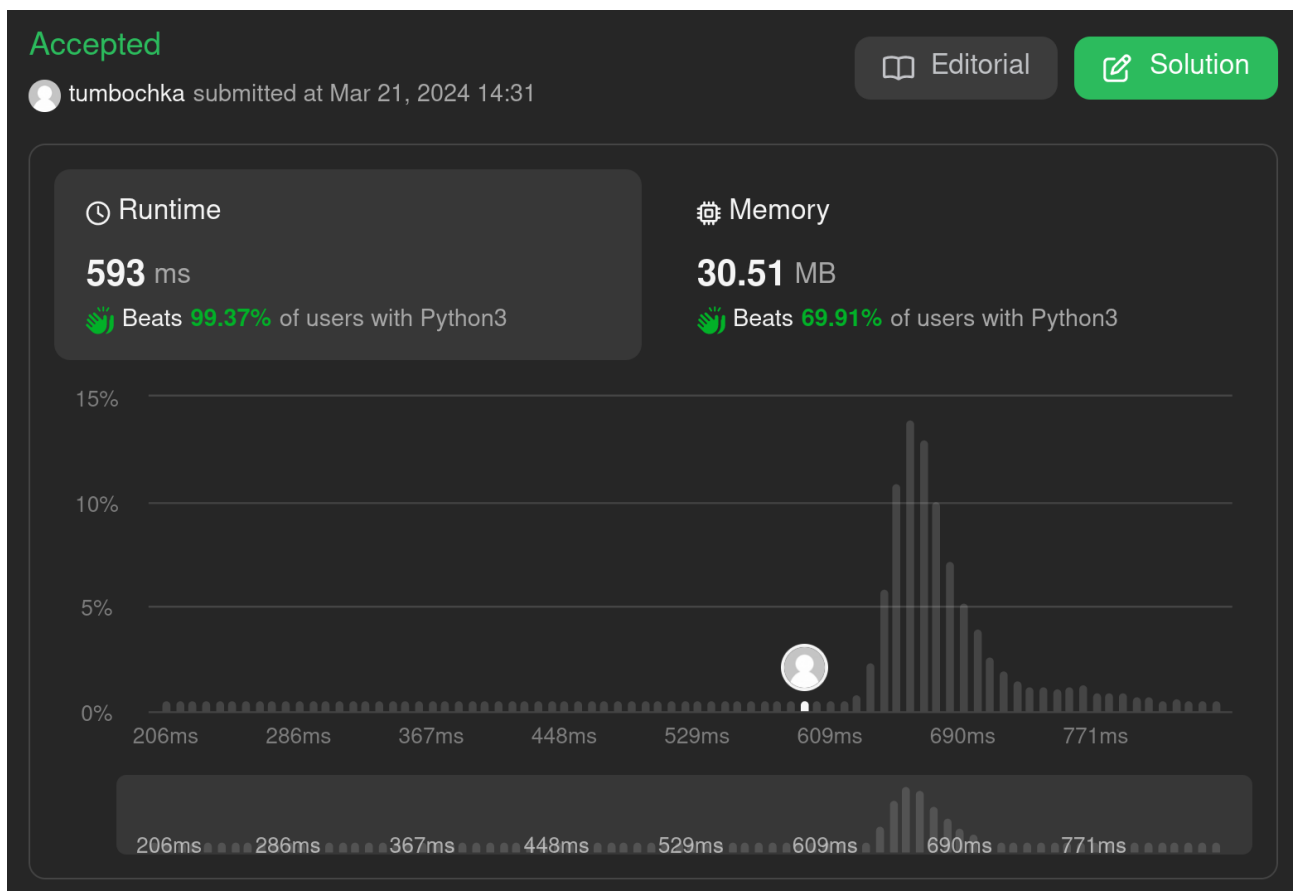


Рисунок 4.3 — Результат решения


4.2 Задача Димы


Была решена задача 42 [10]. Ниже приведены её описание 4.4, решение 4.5 и результат 4.6.

42. Trapping Rain Water

Solved 

Hard

 Topics

 Companies

Given n non-negative integers representing an elevation map where the width of each bar is 1, compute how much water it can trap after raining.

Example 1:



Input: height = [0,1,0,2,1,0,1,3,2,1,2,1]

Output: 6

Explanation: The above elevation map (black section) is represented by array [0,1,0,2,1,0,1,3,2,1,2,1]. In this case, 6 units of rain water (blue section) are being trapped.

Example 2:

Рисунок 4.4 — Описание задачи

```

class Solution:
    def trap(self, height: List[int]) -> int:
        ma = max(height)
        height_sum = sum(height)

        left_ind = 0
        max_left = 0
        while height[left_ind] < ma:
            max_left = max(max_left, height[left_ind])
            height[left_ind] = max_left
            left_ind += 1

        right_ind = len(height) - 1
        max_right = 0
        while height[right_ind] < ma:
            max_right = max(max_right, height[right_ind])
            height[right_ind] = max_right
            right_ind -= 1

        return sum(height[:left_ind] + height[right_ind + 1:]) + (right_ind - left_ind + 1) * ma - height_sum

```

Рисунок 4.5 — Код решения

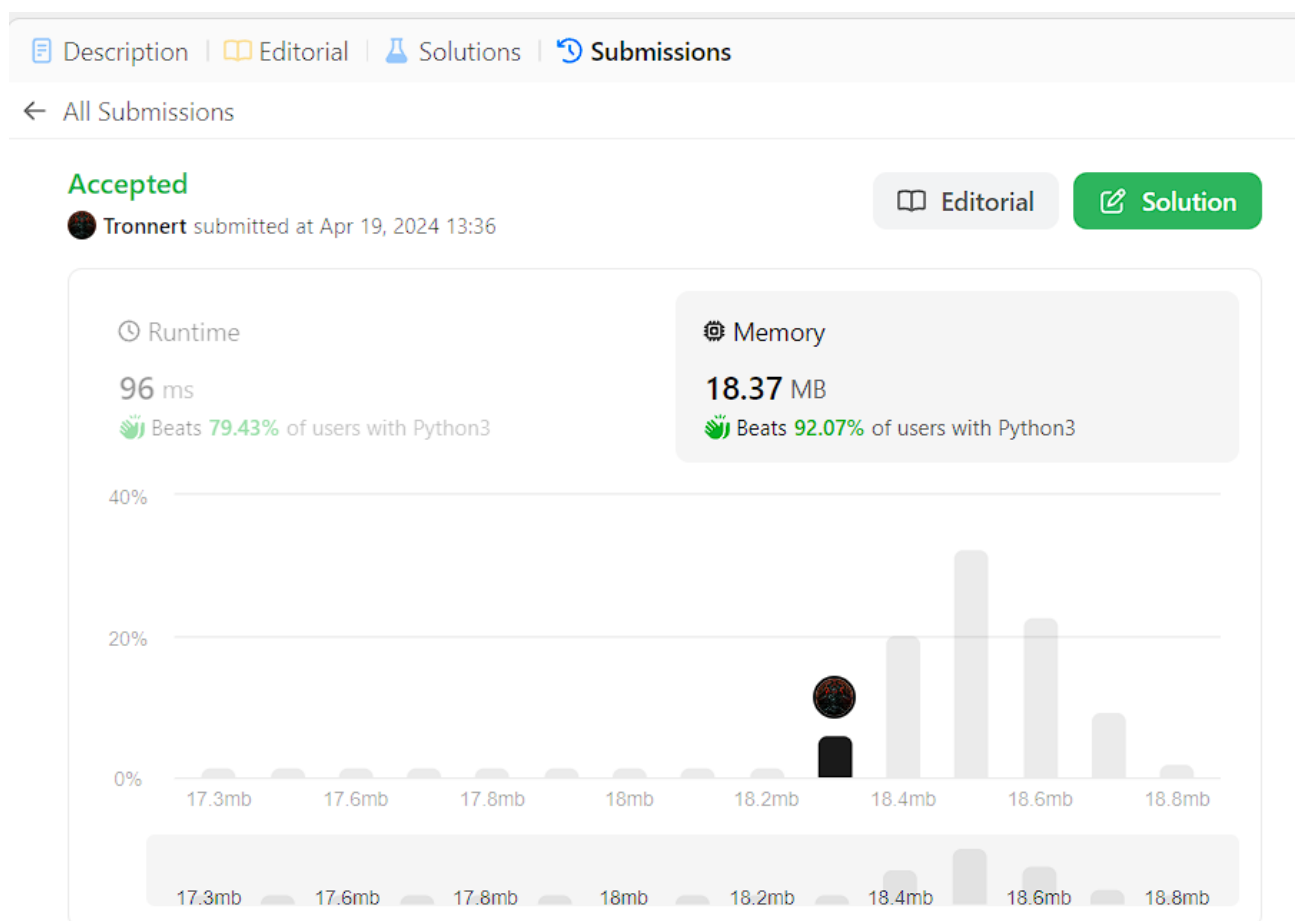


Рисунок 4.6 — Результат решения

4.3 Задача Ивана

Была решена задача 1411 [10]. Ниже приведены её описание 4.7, решение 4.8 и результат 4.9.

1411. Number of Ways to Paint $N \times 3$ Grid

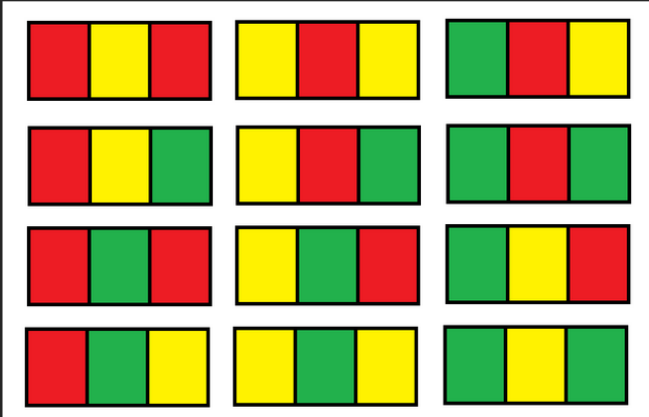
Solved

Hard Topics Companies Hint

You have a `grid` of size $n \times 3$ and you want to paint each cell of the grid with exactly one of the three colors: **Red**, **Yellow**, or **Green** while making sure that no two adjacent cells have the same color (i.e., no two cells that share vertical or horizontal sides have the same color).

Given n the number of rows of the grid, return *the number of ways* you can paint this `grid`. As the answer may grow large, the answer **must be** computed modulo $10^9 + 7$.

Example 1:



Input: $n = 1$
Output: 12
Explanation: There are 12 possible way to paint the grid as shown.

Рисунок 4.7 — Описание задачи

```
</> Code | Code Sample X
Python3 v Auto
1 class Solution:
2     def numOfWays(self, n: int) -> int:
3         aba = 6
4         abc = 6
5         mod = 10**9 + 7
6         for _ in range(n - 1):
7             aba, abc = (3 * aba + 2 * abc) % mod, (2 * aba + 2 * abc) % mod
8         return (aba + abc) % mod
9
```

Рисунок 4.8 — Код решения

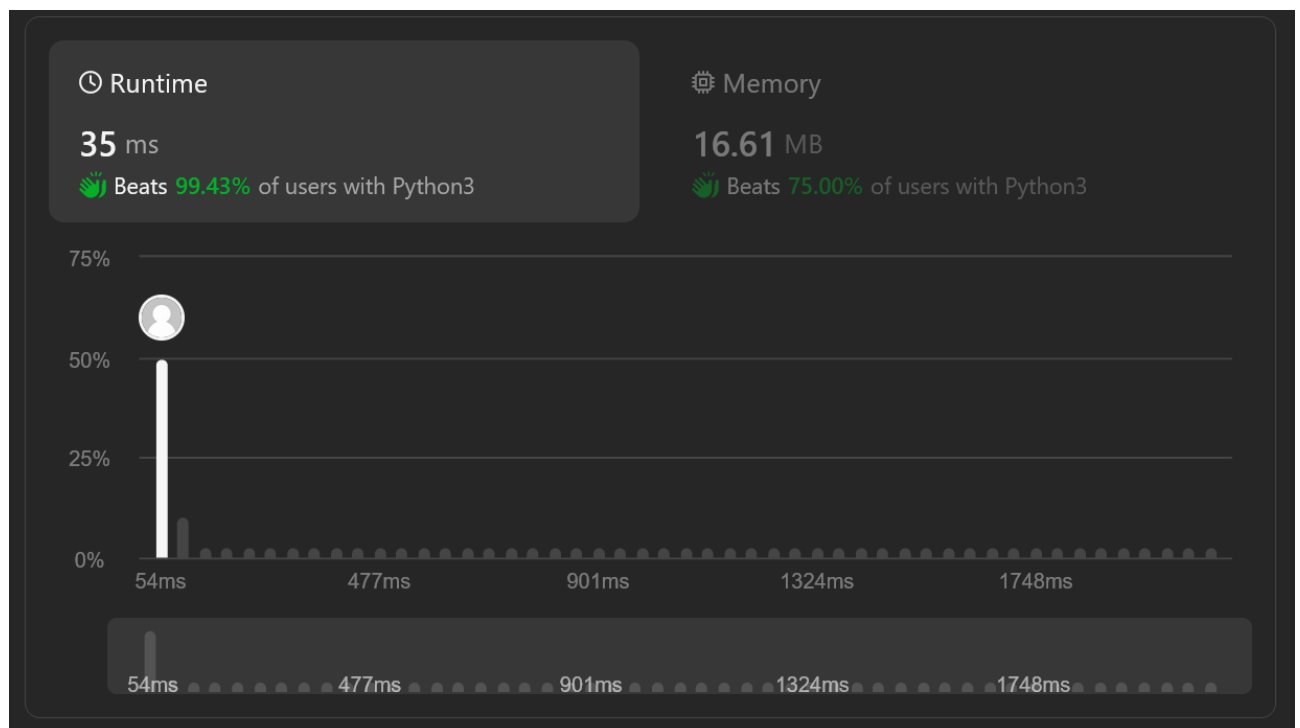


Рисунок 4.9 — Результат решения

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Википедия. [Электронный ресурс]: [сайт]. - URL: <https://wikipedia.org/> (дата обращения: 19.04.2024).
2. Библиотека requests. [Электронный ресурс]: [сайт]. - URL: <https://pypi.org/project/requests/> (дата обращения: 19.04.2024).
3. Документация к модулю asyncio [Электронный ресурс]: [сайт]. - URL: <https://docs.python.org/3/library/asyncio.html> (дата обращения: 19.04.2024).
4. Документация библиотеки python-telegram-bot [Электронный ресурс]: [сайт]. - URL: <https://docs.python-telegram-bot.org/> (дата обращения: 19.04.2024).
5. Yandex Cloud [Электронный ресурс]: [сайт]. - URL: <https://cloud.yandex.ru/> (дата обращения: 19.04.2024).
6. YandexGPT [Электронный ресурс]: [сайт]. - URL: <https://ya.ru/ai/gpt-2> (дата обращения: 19.04.2024).
7. Телеграм-бот ИТМО.Савелий [Электронный ресурс]: [сайт]. - URL: https://t.me/itmo_history_helper_bot (дата обращения: 19.04.2024).
8. Гитхаб репозиторий ИТМО.Савелий [Электронный ресурс]: [сайт]. - URL: https://github.com/koldun256/hist_assistant (дата обращения: 19.04.2024).
9. Leetcode 84 [Электронный ресурс]: [сайт]. - URL: <https://leetcode.com/problems/largest-rectangle-in-histogram> (дата обращения: 19.04.2024).
10. Leetcode 42 [Электронный ресурс]: [сайт]. - URL: <https://leetcode.com/problems/trapping-rain-water/description> (дата обращения: 19.04.2024).

11. Leetcode 1411 [Электронный ресурс]: [сайт]. - URL: <https://leetcode.com/problems/number-of-ways-to-paint-n-3-grid/description/> (дата обращения: 19.04.2024).