

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ  
ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по дополнительным задачам  
по курсу «Алгоритмы и структуры данных»

Выполнил:  
Катков Алексей Сергеевич  
K3239

Проверил:  
Афанасьев А.В

Санкт-Петербург  
2024 г.

## Оглавление

Дополнительные задачи.....	3
7 задача (1 лаб. Работа). Проблема сапожника (0.5 балла) .....	3
17 задача (1 лаб. Работа). Ход конем (2.5 балла) .....	4
16 Задача (2 лаб. Работа). К-й максимум [2 s, 512 Mb, 3 балла] .....	5

## Дополнительные задачи

### 7 задача (1 лаб. Работа). Проблема сапожника (0.5 балла)

Текст задачи:

В некоей воинской части есть сапожник. Рабочий день сапожника длится K минут. Заведующий складом оценивает работу сапожника по количеству починенной обуви, независимо от того, насколько сложный ремонт требовался в каждом случае. Дано n сапог, нуждающихся в починке. Определите, какое максимальное количество из них сапожник сможет починить за один рабочий день

Листинг кода:

```
def max_boots(K, n, repair_times):
    repair_times.sort()
    count = 0
    time_worked = 0

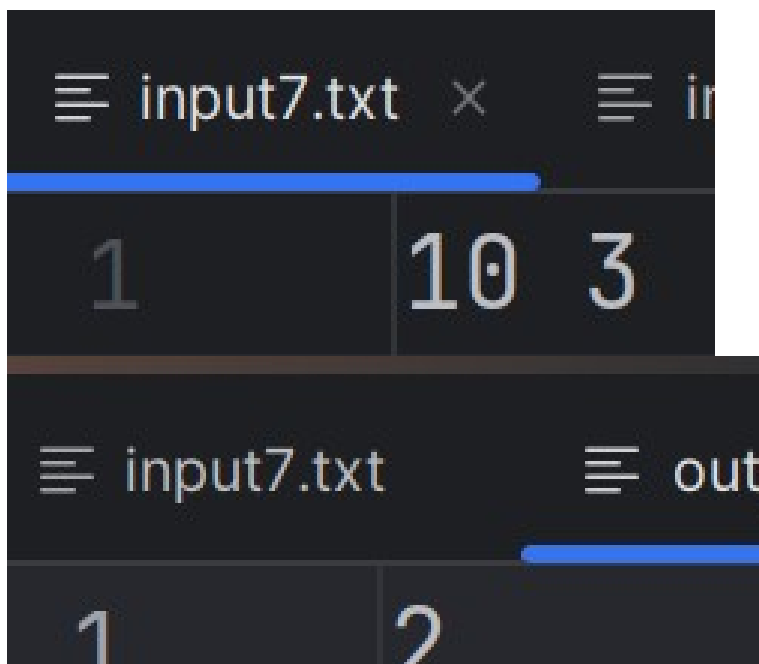
    for time in repair_times:
        if time_worked + time <= K:
            count += 1
            time_worked += time
        else:
            break

    return count

with open("input7.txt", "r") as file:
    K, n = map(int, file.readline().split())
    repair_times = list(map(int, file.readline().split()))

result = max_boots(K, n, repair_times)

with open("output7.txt", "w") as file:
    file.write(str(result))
```



	Время выполнения	Затраты памяти
Нижняя граница диапазона значений входных данных из текста задачи	0.016483710	8.6МБ
Пример из задачи	0.017884987	8.6МБ
Верхняя граница диапазона значений входных данных из текста задачи	0.024995917	10.2МБ

Вывод по задаче: эта задача не вызвала у меня затруднений при решении

## 17 задача (1 лаб. Работа). Ход конем (2.5 балла)

Текст задачи:

Шахматная ассоциация решила оснастить всех своих сотрудников такими телефонными номерами, которые бы набирались на кнопочном телефоне ходом коня. Например, ходом коня набирается телефон 340-49-27. При этом телефонный номер не может начинаться ни с цифры 0, ни с цифры 8. 1 2 3 4 5 6 7 8 9 . 0 . Напишите программу, определяющую количество телефонных номеров длины N, набираемых ходом коня. Поскольку таких номеров может быть очень много, выведите ответ по модулю  $10^9$ .

Листинг кода:

```
def count_phone_numbers(N):
    dp = [[0 for i in range(10)] for i in range(N)]

    for i in range(10):
        if i != 8 and i != 0:
            dp[0][i] = 1

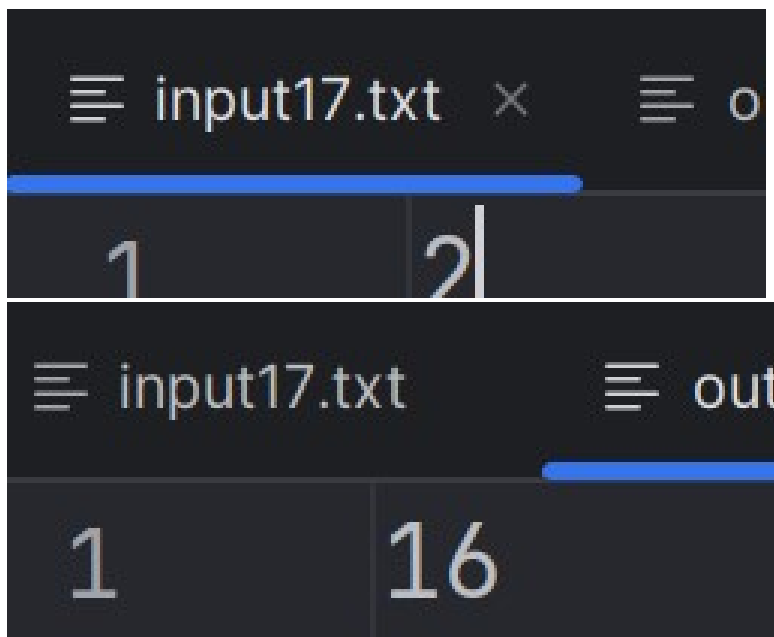
    for i in range(1, N):
        dp[i][0] = dp[i - 1][4] + dp[i - 1][6]
        dp[i][1] = dp[i - 1][6] + dp[i - 1][8]
        dp[i][2] = dp[i - 1][7] + dp[i - 1][9]
        dp[i][3] = dp[i - 1][4] + dp[i - 1][8]
        dp[i][4] = dp[i - 1][3] + dp[i - 1][9] + dp[i - 1][0]
        dp[i][6] = dp[i - 1][1] + dp[i - 1][7] + dp[i - 1][0]
        dp[i][7] = dp[i - 1][2] + dp[i - 1][6]
        dp[i][8] = dp[i - 1][1] + dp[i - 1][3]
        dp[i][9] = dp[i - 1][2] + dp[i - 1][4]

    total = sum(dp[N - 1])
    return total

with open('input17.txt', 'r') as file:
    N = int(file.readline())

result = count_phone_numbers(N)

with open('output17.txt', 'w') as file:
    file.write(str(result))
```



	Время выполнения	Затраты памяти
Нижняя граница диапазона значений входных данных из текста задачи	0.021643474	14.2МБ
Пример из задачи	0.038667102	14.2МБ
Верхняя граница диапазона значений входных данных из текста задачи	0.358977102	25.2МБ

Вывод по задаче: эта задача показалась мне интересной

## 16 Задача (2 лаб. Работа). К-й максимум [2 s, 512 Мб, 3 балла]

Текст задачи: Напишите программу, реализующую структуру данных, позволяющую добавлять и удалять элементы, а также находить k-й максимум.

Листинг кода:

```
class KMaxStructure:
    def __init__(self):
        self.elements = []

    def add_element(self, key):
        self.elements.append(key)

    def delete_element(self, key):
        self.elements.remove(key)

    def find_kth_max(self, k):
        sorted_elements = sorted(self.elements, reverse=True)
        return sorted_elements[k-1]

with open("input16.txt", "r") as file:
    n = int(file.readline().strip())
    commands = [list(map(int, file.readline().strip().split())) for _ in range(n)]

kmax_structure = KMaxStructure()
result = []

for command in commands:
    if command[0] == 1:
        kmax_structure.add_element(command[1])
    elif command[0] == 0:
        result.append(str(kmax_structure.find_kth_max(command[1])))
    elif command[0] == -1:
        kmax_structure.delete_element(command[1])

with open("output16.txt", "w") as file:
    file.write("\n".join(result))
```

≡ input16.txt × ≡

1	11	
2	+1	5
3	+1	3
4	+1	7
5	0	1
6	0	2
7	0	3
8	-1	5

input16.txt	output16.txt
1	7
2	5
3	3
4	10

	Время выполнения	Затраты памяти
Нижняя граница диапазона значений входных данных из текста задачи	0.059576120	33.2МБ
Пример из задачи	0.060067122	33.2МБ
Верхняя граница диапазона значений входных данных из текста задачи	0.692168763	72.1МБ

Вывод по задаче: в рамках этой задачи я реализовал подобие своей структуры данных