САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по лабораторной работе №7

по курсу «Алгоритмы и структуры данных»

Тема: Динамическое программирование

Вариант 20

Выполнил:

Смирнов Георгий Валерьевич

К3139

Проверил:

Афанасьев А.В.

Санкт-Петербург

2024 г.

# Содержание отчета

[**Содержание отчета 2**](#_gjdgxs)

[**Задачи по варианту 3**](#_30j0zll)

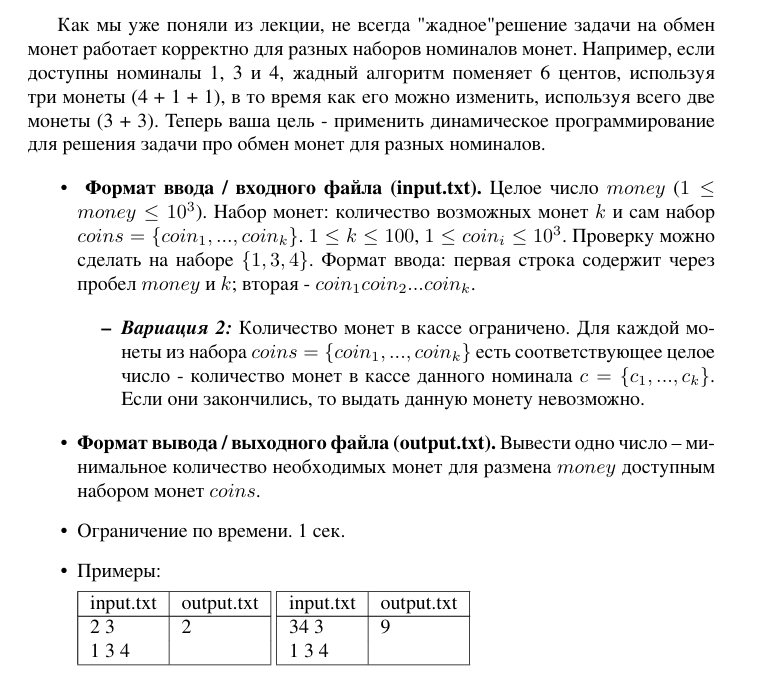
[Задача №1. Обмен монет 3](#_1fob9te)

[Задача №6. Наибольшая возрастающая последовательность 6](#_3znysh7)

[**Вывод 23**](#_1t3h5sf)

# Задачи по варианту

## Задача №1. Обмен монет.



**Код программы**

def min\_coins(money, coins):

# Инициализация массива для хранения минимального количества монет

dp = [float('inf')] \* (money + 1)

dp[0] = 0

# Заполнение массива dp

for coin in coins:

for x in range(coin, money + 1):

dp[x] = min(dp[x], dp[x - coin] + 1)

return dp[money] if dp[money] != float('inf') else -1

def process\_input(input\_file, output\_file):

with open(input\_file, 'r') as infile:

money, k = map(int, infile.readline().strip().split())

coins = list(map(int, infile.readline().strip().split()))

result = min\_coins(money, coins)

with open(output\_file, 'w') as outfile:

outfile.write(f"{result}\n")

def main():

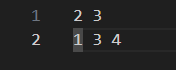
process\_input('input.txt', 'output.txt')

if \_\_name\_\_ == "\_\_main\_\_":

main()

Результат работы кода на примерах из текста задачи:

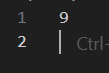
1)





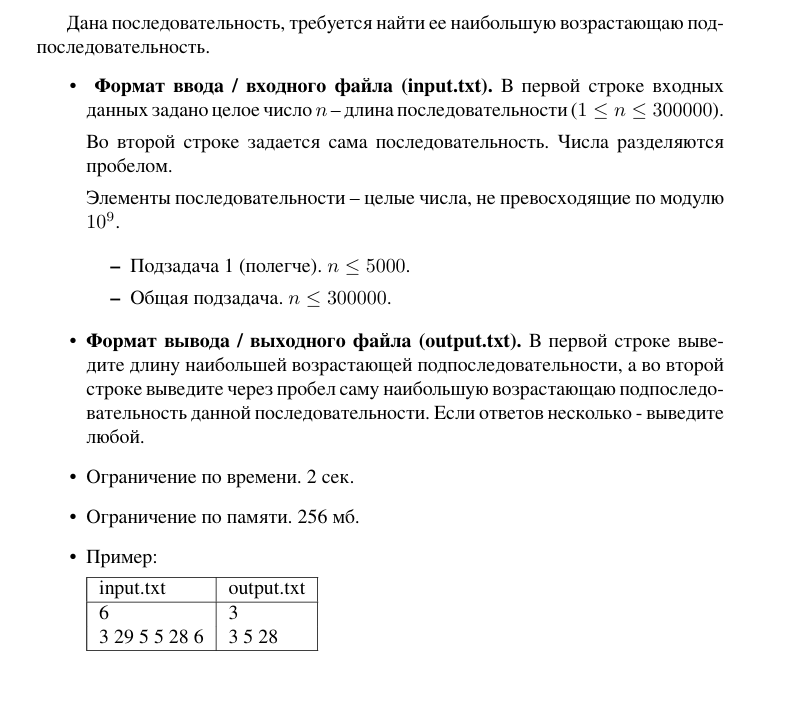
2)





|  | время выполнения, с |
| --- | --- |
| пример из варианта | 0.000487400 |
|  | 0.003130700 |

## Задача №6. [Наибольшая возрастающая последовательность](#_3znysh7).



**Код программы**

class ThreadedMap:

def \_\_init\_\_(self):

# Инициализация словаря для хранения ключ-значение и списка для порядка вставки

self.map = {}

self.order = []

def put(self, x, y):

# Добавление или обновление значения

if x not in self.map:

self.order.append(x)

self.map[x] = y

def delete(self, x):

# Удаление ключа, если он существует

if x in self.map:

del self.map[x]

self.order.remove(x)

def get(self, x):

# Получение значения по ключу или 'none', если ключа нет

return self.map.get(x, 'none')

def prev(self, x):

# Получение предыдущего значения в порядке вставки

if x not in self.map:

return 'none'

index = self.order.index(x)

if index == 0:

return 'none'

return self.map[self.order[index - 1]]

def next(self, x):

# Получение следующего значения в порядке вставки

if x not in self.map:

return 'none'

index = self.order.index(x)

if index == len(self.order) - 1:

return 'none'

return self.map[self.order[index + 1]]

def process\_operations(input\_file, output\_file):

# Обработка операций из входного файла и запись результатов в выходной файл

threaded\_map = ThreadedMap()

with open(input\_file, 'r') as infile, open(output\_file, 'w') as outfile:

n = int(infile.readline().strip())

for \_ in range(n):

operation = infile.readline().strip().split()

op\_type = operation[0]

x = operation[1]

if op\_type == 'put':

y = operation[2]

threaded\_map.put(x, y)

elif op\_type == 'delete':

threaded\_map.delete(x)

elif op\_type == 'get':

result = threaded\_map.get(x)

outfile.write(f"{result}\n")

elif op\_type == 'prev':

result = threaded\_map.prev(x)

outfile.write(f"{result}\n")

elif op\_type == 'next':

result = threaded\_map.next(x)

outfile.write(f"{result}\n")

def main():

# Основная функция для запуска обработки операций

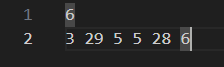
process\_operations('input.txt', 'output.txt')

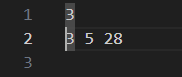
if \_\_name\_\_ == "\_\_main\_\_":

main()

Результат работы кода на примерах из текста задачи:

1)





|  | время, с | память, MB |
| --- | --- | --- |
| пример | 0.003274 | 0.017998 |