

Московский авиационный институт  
(национальный исследовательский университет)

Факультет информационных технологий и прикладной  
математики

Кафедра вычислительной математики и программирования

Лабораторная работа №8 по курсу «Дискретный анализ»

Студент: Г. С. Шубин  
Преподаватель: Н. С. Капралов  
Группа: М8О-308Б  
Дата:  
Оценка:  
Подпись:

Москва, 2022

## Лабораторная работа №8

### Задча:

Разработать жадный алгоритм решения задачи, определяемой своим вариантом. Доказать его корректность, оценить скорость и объём затрачиваемой оперативной памяти.

Реализовать программу на языке C или C++, соответствующую построенному алгоритму. Формат входных и выходных данных описан в варианте задания.

### Вариант №1:

На первой строке заданы два числа,  $N$  и  $p > 1$ , определяющие набор монет некоторой страны с номиналами  $p_0, p_1, \dots, p_{N-1}$ . Нужно определить наименьшее количество монет, которое можно использовать для того, чтобы разменять заданную на второй строчке сумму денег  $M \leq 2^{32} - 1$  и распечатать для каждого  $i$ -го номинала на  $i$ -ой строчке количество участвующих в размене монет. Кроме того, нужно обосновать почему жадный выбор неприменим в общем случае (когда номиналы могут быть любыми) и предложить алгоритм, работающий при любых входных данных.

### Формат входных данных:

На первой строке заданы два числа  $N$  и  $p > 1$ , определяющие набор монет некоторой страны с номиналами  $p_0, p_1, \dots, p_{N-1}$ . На второй строчке находится сумма денег которую необходимо разменять  $M \leq 2^{32} - 1$ .

### Формат результата:

Для каждого  $i$ -го номинала на  $i$ -ой строчке вывести количество участвующих в размене монет.

# 1 Описание

Основная идея жадного алгоритма в том, чтобы на каждом этапе решения подзадачи выбирать локально оптимальное решение и в итоге получить оптимальное решение для всей задачи.

До тех пор, пока заданная сумма больше нуля, из нее вычитается максимально возможное число монет так: сначала это монеты наибольшего номинала, потом номинал меньше и т. д. То есть, если монеты определенного номинала можно вычесть, то вычитаем максимальное возможное количество, иначе переходим к следующему по убыванию номиналу.

Такое решение является оптимальным, так как номиналы представляют собой степени какого-либо числа, следовательно монету одного номинала можно заменить несколькими монетами большего номинала, если это не 1.

Если же номиналы не будут представлять собой степени какого-либо числа, тогда данный алгоритм не будет оптимальным. В таком случае необходимо воспользоваться динамическим программированием. Суть алгоритма в том, чтобы в массиве запоминать минимальное количество монет, которое нужно для размена от 1 до необходимой суммы монет.

Сложность используемого жадного алгоритма: линейная.

## 2 Исходный код

```
1 | #include <iostream>
2 | #include <vector>
3 |
4 | int main()
5 | {
6 |     int N, p;
7 |     unsigned int M;
8 |     std::cin >> N >> p >> M;
9 |     int k = N - 1;
10 |    std::vector<int> p_arr(N);
11 |    std::vector<int> res_arr(N);
12 |    int current = 1;
13 |    for (int i = 0; i < N; ++i) {
14 |        p_arr[i] = current;
15 |        current = current * p;
16 |    }
17 |    while (M > 0) {
18 |        int count = M / p_arr[k];
19 |        if (count == 0) {
20 |            k--;
21 |        }
22 |        else {
23 |            M = M - count * p_arr[k];
24 |            res_arr[k] = count;
25 |            k--;
26 |        }
27 |    }
28 |    for (int i = 0; i < N; ++i) {
29 |        std::cout << res_arr[i] << '\n';
30 |    }
31 |    return 0;
32 | }
```

### 3 Консоль

```
gregory@DESKTOP-7L8SUG4:~/DA/Lab8$ ./solution
3 5
71
1
4
2
```

```
gregory@DESKTOP-7L8SUG4:~/DA/Lab8$ ./solution
4 4
108
0
3
2
1
```

```
gregory@DESKTOP-7L8SUG4:~/DA/Lab8$ ./solution
2 17
105
3
6
```

```
gregory@DESKTOP-7L8SUG4:~/DA/Lab8$ ./solution
1 1
15
15
```

## 4 Тест производительности

Номиналы	Время(сек.)
100	0,000014
1000	0,000136
10000	0,001302
100000	0,01245
1000000	0,1388

В таблице представлены результаты выполнения жадного алгоритма. Результаты выполнения наивного алгоритма не приведены, поскольку время переваливало за целую часть секунд на тестах с низким номиналом.

## 5 Выводы

Выполняя данную лабораторную работу, я познакомился с жадными алгоритмами. Основное отличие жадных алгоритмов от динамического программирования в том, что жадные алгоритмы не перебирают все возможные варианты решения подзадач, в поисках оптимального, а сразу берут наилучшее решение, которое заранее определено оптимальным. Такой подход сокращает время работы программы, но при этом применение жадного алгоритма должно учитываться еще на этапе планирования решения, так как подходит к более узкому множеству задач.

## Список литературы

- [1] Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. *Алгоритмы: построение и анализ, 2-е издание.* — Издательский дом «Вильямс», 2007. Перевод с английского: И. В. Красиков, Н. А. Орехова, В. Н. Романов. — 1296 с. (ISBN 5-8459-0857-4 (рус.))