

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: А. А. Терво  
Преподаватель: А. А. Кухтичев  
Группа: М8О-207Б  
Дата:  
Оценка:  
Подпись:

Москва, 2022

## Лабораторная работа №8

**Задача:** 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.

На координатной прямой даны несколько отрезков с координатами  $[L_i, R_i]$ . Необходимо выбрать минимальное количество отрезков, которые бы полностью покрыли интервал  $[0, M]$ .

**Формат входных данных:** На первой строке располагается число  $N$ , за которым следует  $N$  строк на каждой из которой находится пара чисел  $L_i, R_i$ ; последняя строка содержит в себе число  $M$ .

**Формат результата:** На первой строке число  $K$  выбранных отрезков, за которым следует  $K$  строк, содержащих в себе выбранные отрезки в том же порядке, в котом они встретились во входных данных. Если покрыть интервал невозможно, нужно распечатать число 0.

# 1 Описание

Требуется решить задачу при помощи жадного алгоритма. В [1] сказано, что суть алгоритма состоит в принятии локально оптимальных решений на каждом этапе, допуская, что конечное решение также окажется оптимальным.

Жадный алгоритм можно использовать не всегда, у него есть условия применимости:

- Применим принцип жадного выбора
- Оптимальность для подзадач

Говорят, что к оптимизационной задаче применим принцип жадного выбора, если последовательность локально оптимальных выборов даёт глобально оптимальное решение.

Говорят, что задача обладает свойством оптимальности для подзадач, если оптимальное решение задачи содержит в себе оптимальные решения для всех её подзадач.

## 2 Исходный код

Решение содержит следующие шаги:

- Записываем отрезки в вектор
- Сортируем вектор
- Обходим его, проверяя каждую тройку отрезков, сохраняя максимальную площадь и соответствующие ей отрезки

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  #include <math.h>
5
6  int main () {
7      int n;
8      std::cin >> n;
9      long long int a = 0, b = 0, c = 0;
10     long long int maxA = 0, maxB = 0, maxC = 0;
11     double maxS = -1;
12     std::vector <long long int> lines(n);
13     for (int i = 0; i < n; i++) {
14         std::cin >> lines[i];
15     }
16     if (n >= 3) {
17         std::sort(lines.begin(), lines.end());
18         for (int i = n - 1; i >= 2; i--) {
19             if (lines[i] < lines[i - 1] + lines[i - 2]) {
20                 a = lines[i];
21                 b = lines[i - 1];
22                 c = lines[i - 2];
23             }
24             if ((a != 0) && (b != 0) && (c != 0) && (a < b + c)) {
25                 double p = double(a + b + c) / 2;
26                 double s = double(sqrt(p * (p - a) * (p - b) * (p - c)));
27                 if (s > maxS) {
28                     maxS = s;
29                     maxA = a;
30                     maxB = b;
31                     maxC = c;
32                 }
33             }
34         }
35     } else {
36         std::cout << 0 << std::endl;
37     }
```

```

38     if (maxS > 0) {
39         printf (".3lf\n", maxS);
40         std::cout << maxC << " " << maxB << " " << maxA << std::endl;
41     }
42     else {
43         std::cout << 0 << std::endl;
44     }
45
46     return 0;
47 }

```

### 3 Консоль

```
[alext@alext-pc solution]$ ./solution
6
6
5
4
3
2
1
9.922
6 5 4
[alext@alext-pc solution]$ ./solution
4
1
2
3
5
0
```

## 4 Выводы

Во время выполнения этой лабораторной работы я изучил, что такое жадные алгоритмы, и где они применяются.

Однако, мне кажется, что решение моей задачи с трудом можно назвать жадным алгоритмом. Так как мы не можем знать, какой отрезок будет выгодно выбрать следующим, нам приходится просто перебирать возможные комбинации.

Тем не менее, решение выполняется за  $O(n * \log(n) + n)$ , то есть сортировка плюс обход вектора с отрезками.

## Список литературы

[1] *Жадный алгоритм* — Википедия.

URL: [http://ru.wikipedia.org/wiki/Жадный\\_алгоритм](http://ru.wikipedia.org/wiki/Жадный_алгоритм) (дата обращения: 16.12.2022).

[2] *Жадные алгоритмы* / Хабр.

URL: <https://habr.com/ru/post/120343/> (дата обращения: 16.12.2022).