

**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: Д. М. Чистяков
Преподаватель: А. А. Кухтичев
Группа: М8О-306Б-20
Дата:
Оценка:
Подпись:

Москва, 2023

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.

Заданы длины N отрезков, необходимо выбрать три таких отрезка, которые образовывали бы треугольник с максимальной площадью.

Формат входных данных: На первой строке находится число N , за которым следует N строк с целыми числами-длинами отрезков.

Формат результата: Если никакого треугольника из заданных отрезков составить нельзя — 0, в противном случае на первой строке площадь треугольника с тремя знаками после запятой, на второй строке — длины трёх отрезков, составляющих этот треугольник. Длины должны быть отсортированы.

1 Описание

Данный жадный алгоритм основан на расположении длин сторон в порядке убывания и проверке начиная сверху, берется три самых больших стороны считается площадь, если такой треугольник возможен, делаем проверку сравнивая с текущей наибольшей площадью, если значение больше, то запоминаем. Из-за сортировки требуется $O(n \log n)$ времени.

2 Исходный код

Код: main.cpp

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  #include <cmath>
5
6  double Area(int s1, int s2, int s3) {
7      double p = (s1 + s2 + s3) * 0.5;
8      return sqrt(p * (p - s1) * (p - s2) * (p - s3));
9  }
10
11 bool ValidTriangle(int s1, int s2, int s3) {
12     return (s1 < (s2 + s3)) && (s2 < (s1 + s3)) && (s3 < (s1 + s2));
13 }
14
15 int main() {
16     std::vector<int> data;
17     int n = 0;
18     int s = 0;
19     int s1 = 0;
20     int s2 = 0;
21     int s3 = 0;
22     double max_area = 0.0;
23     double cur_area = 0.0;
24
25     std::cin >> n;
26     for (int i = 0; i < n; ++i) {
27         std::cin >> s;
28         data.push_back(s);
29     }
30
31     std::sort(data.begin(), data.end(), std::greater<int>());
32
33     for (int i = 1; i < int(data.size() - 1); ++i) {
34         if (data.size() < 3) {
35             break;
36         }
37         if (ValidTriangle(data[i - 1], data[i], data[i + 1])) {
38             cur_area = Area(data[i - 1], data[i], data[i + 1]);
39             if (cur_area > max_area) {
40                 max_area = cur_area;
41                 s1 = data[i + 1];
42                 s2 = data[i];
43                 s3 = data[i - 1];
44             }
45         }
46     }
```

```

47 |
48 |     if (max_area == 0.0) {
49 |         std::cout << 0 << '\n';
50 |     }
51 |     else {
52 |         printf("%.3f\n", max_area);
53 |         std::cout << s1 << ' ' << s2 << ' ' << s3 << '\n';
54 |     }
55 |     return 0;
56 | }

```

3 Консоль

```

den@vbox:~/Документы/DA/lab8$ ./a.out
4
1 2 3 5
0

```

4 Тест производительности

Сравним реализованный алгоритм с наивным алгоритмом, который не всегда даёт верный ответ. Тест состоит из нахождения наибольшей площади для 50000 и 100000 сторон

Моя реализация:

```

den@vbox:~/Документы/DA/lab8$ ./benchmark
Time for algo with 50000 sides: 0.12 seconds
Time for algo with 100000 sides: 0.29 seconds

```

Видно, что алгоритм работает, явно лучше чем наивный алгоритм за $O(n^2)$.

5 Выводы

Выполнив восьмую лабораторную работу по курсу «Дискретный анализ», я познакомился с жадными алгоритмами. Изучил классические задачи и их методы решения, которые можно решать данным видом алгоритмов, написал простой жадный алгоритм по определению наибольшей площади треугольника.

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.))