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. С. Пивницкий
Преподаватель: А. А. Кухтичев
Группа: М8О-206Б-19

Дата: 01.01.2021
Оценка:
Подпись:

Москва, 2021

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++, соответсвующую 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 требует $O(nlog_n)$ времени.

2 Исходный код

Код: main.cpp

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <cmath>
#include <iomanip>

bool CompareFunc(int const& lhs, int const& rhs) { return lhs > rhs; }

bool ValidTriangle(int const& s_1, int const& s_2, int const& s_3)
{
if((s_1 < (s_2 + s_3)) && (s_2 < (s_1 + s_3)) && (s_3 < (s_1 + s_2)))
return true;
else
return false;
}

double Area(int const& s_1, int const& s_2, int const& s_3)
{
double p = 0.5 * (s_1 + s_2 + s_3);
return sqrt(p) * sqrt(p - s_1) * sqrt(p - s_2) * sqrt(p - s_3);
}

int main()
{
std::vector<int> data;
int n = 0, s = 0, s_1 = 0, s_2 = 0, s_3 = 0;
double max_area = 0.0, cur_area = 0.0;

std::cin >> n;
for (int i = 0; i < n; ++i)
{
std::cin >> s;
data.push_back(s);
}

std::sort(data.begin(), data.end(), CompareFunc);

for(int i = 1; i < int(data.size() - 1); ++i)
{
if(data.size() < 3)
break;
if(ValidTriangle(data[i - 1], data[i], data[i + 1]))
{
cur_area = Area(data[i - 1], data[i], data[i + 1]);
if(cur_area > max_area)
{
```

```
47 max_area = cur_area;
48 s_1 = data[i + 1];
49 s_2 = data[i];
50 s_3 = data[i - 1];
51 }
52 }
53 }
54
55 if(max_area == 0)
56 std::cout << 0 << '\n';
57 else
58 {
59 printf("%.3f\n", max_area);
60 std::cout << s_1 << ' ' << s_2 << ' ' << s_3 << '\n';
61 }
62 return 0;
63 }
```

3 Консоль

```
(py37) ~ /DA_labs/lab8$ make
g++ -g -O2 -pedantic -std=c++17 -Wall -Wextra -Werror main.cpp -o solution
(py37) ~ /DA_labs/lab8$ cat test.txt
4
1
2
3
5
(py37) ~ /DA_labs/lab8$ ./solution <test.txt
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:

```
(py37) ~ /DA_labs/lab7$ make
g++ -g -O2 -pedantic -std=c++17 -Wall -Wextra -Werror main.cpp -o solution
(py37) ~ /DA_labs/lab8$ make bench
g++ -g -O2 -pedantic -std=c++17 -Wall -Wextra -Werror benchmark.cpp -o benchmark
(py37) ~ /DA_labs/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.))