

Лабораторная работа № 8 по курсу дискретного анализа: Жадные алгоритмы

Выполнил студент группы 08-208 МАИ *Левитанов Денис*.

Условие

1. 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.

2. Максимальный треугольник

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 выходных данных: 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.

Метод решения

Считывается количество элементов, и создается вектор для них. Вектор сортируется в порядке убывания. Затем начиная с самых больших элементов берутся три элемента и проверяются на возможность составить из них треугольник (каждая сторона должна быть меньше чем сумма других двух), если это условие выполняется то высчитывается площадь по формуле Герона и выводится ответ. Общее время работы $O(n \log n + n)$ $n \log n$ для сортировки и n для прохода по отсортированной последовательности.

Описание программы

1. main.cpp(Считывание данных, управление программой):

size_t n - количество элементов

vector<int64_t> vec - вектор элементов

double square - площадь

Дневник отладки

Программа не проходила 5 тест из-за того, что не обрабатывалась последняя тройка элементов в массиве.

Тест производительности

1. Time of working 1000: 0.002
2. Time of working 10000: 0.019
3. Time of working 100000: 0.168
4. Time of working 1000000: 1.774

Выводы

Решение с помощью жадных алгоритмов заключается в том, что на каждом этапе решения выбирается оптимальное, в отличие от динамического программирования, где осуществляется просмотр всех возможных вариантов решений. Тем самым можно значительно улучшить асимптотику многих задач. Кроме того, практически все задачи, решаемые жадными алгоритмами можно решить с помощью динамического программирования, но не наоборот.