

Отчёт по лабораторному заданию №2. Построение и анализ выпуклых оболочек

Ю. Н. Лукашкина

julialukashkina@gmail.ru

МГУ имени М. В. Ломоносова, Москва

19 ноября 2016 г.

Аннотация

Данный документ содержит отчет по лабораторной работе №2 курса «Алгоритмика» 617 группы ВМК МГУ.

1 Постановка задачи

Глубиной $D(p)$ точки p в конечном множестве S из n точек на евклидовой плоскости называется число выпуклых оболочек (выпуклых слоев), которые должны быть удалены из S прежде, чем будет удалена точка p .

Пусть $S_m = \{p : D(p) = m, p \in S\}$ — множество точек глубины m . Функцией глубин S называется $F(m) = |S_m|, m = 0, 1, \dots, M(S)$ — количество точек в S , имеющих глубину m .

Глубиной $M(S)$ множества S называется максимум глубины точек, входящих в S .

Требуется реализовать алгоритм вычисления для заданного множества S глубины $M(S)$ и функции глубин $F(m)$.

2 Описание данных

Исходные данные задаются в текстовом файле. Первая запись — число точек, далее координаты точек. Координаты точек — действительные числа в диапазоне $[0, 10^{17}]$. Максимальное число точек $n = 10^6$.

Выходные данные программы: имя файла исходных данных, значение глубины $M(S)$ и таблица значений функции глубин в формате $(m, F(m)), m = 0, 1, \dots, M(S)$.

3 Описание метода решения

Для построения выпуклой оболочки множества в данной работе был исследован и реализован алгоритм Эндрю (Грэхэма–Эндрю). Все точки были отсортированы в лексикографическом порядке (сложность $O(n \log n)$). Далее были построены нижняя и верхняя оболочки множества. Рассмотрим более подробно алгоритм построения нижней оболочки множества, верхняя строится аналогично. Считаем, что на вход функции построения выпуклой оболочки подаются уже отсортированные точки. Первая точка множества (самая левая) в любом случае попадает в выпуклую оболочку множества. Далее рассматриваются все остальные точки множества, на каждом шаге рассматриваются текущая точка и две последние точки выпуклой оболочки. Пока эти точки не образуют правый поворот, последняя точка выпуклой оболочки удаляется. Если описанные выше точки образуют правый поворот, то рассматриваемая точка добавляется в выпуклую оболочку, переходим на следующий шаг к рассмотрению следующей точки из исходного множества. Сложность построения выпуклой оболочки $O(n)$.

Для вычисления глубины множества и построения функции глубины будем итеративно строить выпуклую оболочку множества и удалять точки, ей принадлежащие. Таких итераций в худшем случае будет $O(n)$.

4 Описание программной реализации

В качестве интерфейса программы было выбрано консольное приложение. Приложение принимает на вход один аргумент — абсолютный путь до входного файла. Результатом работы программы является вывод в консоль названия входного файла, глубины множества и функции глубины.

Пример запуска приложения:

```
ConvexHull.exe Q:\mmp_2016\algorithms\Task1\alg_task1.git\Examples\4node.txt
```

Таблица 1. Результаты работы

Название датасета	Время работы (с)	M(S)
4node	0	1
12node	0	1
633node	0.006	36
10000node	0.437	222
100000node	27.482	1045
птички1	0.023	47
рыбки1	0.007	23

Таблица 2. Функция глубины на датасете 4node

m	F(m)
0	3
1	1

Таблица 3. Функция глубины на датасете 12node

m	F(m)
0	7
1	5

В качестве алгоритма сортировки была выбрана сортировка слиянием (класс MergeSorter). Построение выпуклой оболочки реализовано в функции `get_convex_hull`. Все точки, составляющие выпуклую оболочку, помечаются флагом `is_convex_hull`. Удаление точек выпуклой оболочки из множества производится линейным проходом по множеству и проверкой этого флага.

5 Эксперименты

В таблице 1 приведены результаты работы на тестовых наборах данных. По таблице видно, что время работы программной реализации удовлетворяет аналитическим оценкам. В таблицах 2–4 приведены функции глубин на некоторых из этих наборов.

6 Выводы

В данном практическом задании были исследованы алгоритмы построения выпуклых оболочек, для программной реализации был выбран и реализован алгоритм Эндрю. Практически были проверены теоретические оценки сложности алгоритма.

Таблица 4. Функция глубины на датасете 633

m	F(m)
0	16
1	18
2	26
3	21
4	27
5	27
6	19
7	23
8	20
9	21
10	21
11	24
12	23
13	25
14	19
15	22
16	23
17	18
18	18
19	20
20	15
21	19
22	16
23	16
24	15
25	12
26	19
27	15
28	10
29	9
30	9
31	12
32	13
33	9
34	7
35	5
36	1