Вычислительная геометрия: иллюстрации

Кира Вяткина

Санкт-Петербургский государственный университет

6 марта 2011г.

План доклада

1. Выпуклые оболочки
2. Минимальная описанная окружность
3. Треугольник минимальной площади

Определение

- Пусть $S \subset \mathbb{R}^2$ – некоторое подмножество плоскости

Определение

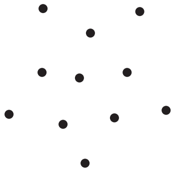
- Пусть $S \subset \mathbb{R}^2$ – некоторое подмножество плоскости
- *Выпуклая оболочка* $\mathcal{CH}(S)$ множества S – наименьшее выпуклое множество, содержащее S

Определение

- Пусть $S \subset \mathbb{R}^2$ – некоторое подмножество плоскости
- *Выпуклая оболочка* $\mathcal{CH}(S)$ множества S – наименьшее выпуклое множество, содержащее S
- Или: $\mathcal{CH}(S)$ – пересечение всех выпуклых множеств, содержащих S

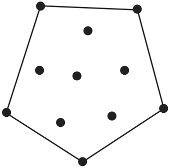
Для множества точек

- Пусть P – множество из n точек на плоскости



Для множества точек

- Пусть P – множество из n точек на плоскости
- $\mathcal{CH}(P)$ – выпуклый многоугольник с вершинами в некоторых точках из S



Сложность построения

- Модель вычислений: вещественнозначная РАМ
    - равнодоступная адресная машина

Сложность построения

- Модель вычислений: вещественнозначная РАМ
  - равнодоступная адресная машина
- Элементарные операции:
  - арифметические операции
  - операции сравнения
  - косвенная адресация памяти

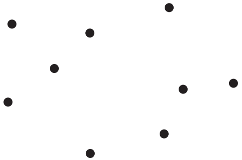
Сложность построения

- Модель вычислений: вещественнозначная РАМ
  - равнодоступная адресная машина
- Элементарные операции:
  - арифметические операции
  - операции сравнения
  - косвенная адресация памяти
  - аналитические функции

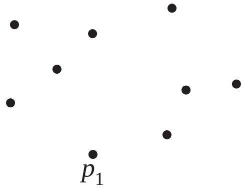
Сложность построения

- Модель вычислений: вещественнозначная РАМ
    - равнодоступная адресная машина
- Элементарные операции:
    - арифметические операции
    - операции сравнения
    - косвенная адресация памяти
    - аналитические функции
- Нижняя оценка сложности построения выпуклой оболочки: $\Omega(n \log n)$

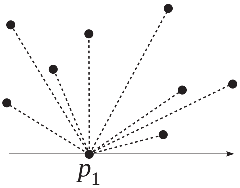
Метод обхода Грэхема



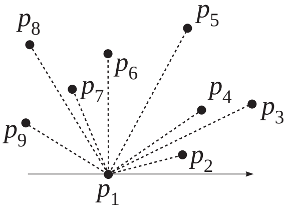
Метод обхода Грэхема



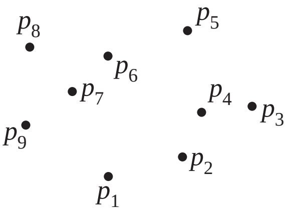
Метод обхода Грэхема



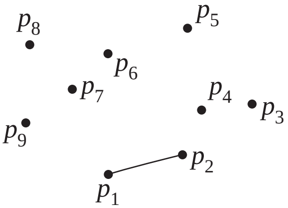
Метод обхода Грэхема



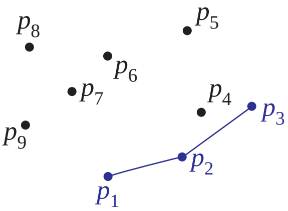
Метод обхода Грэхема



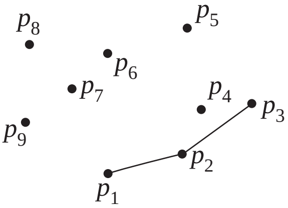
Метод обхода Грэхема



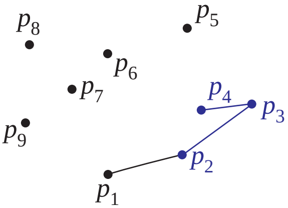
Метод обхода Грэхема



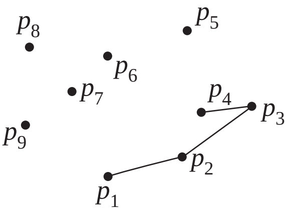
Метод обхода Грэхема



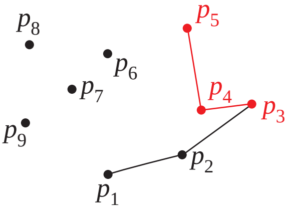
Метод обхода Грэхема



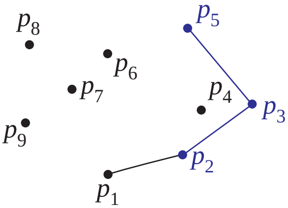
Метод обхода Грэхема



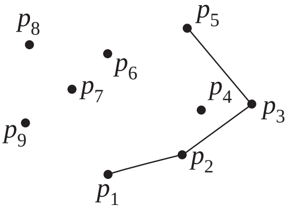
Метод обхода Грэхема



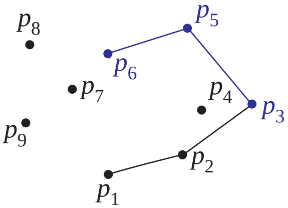
Метод обхода Грэхема



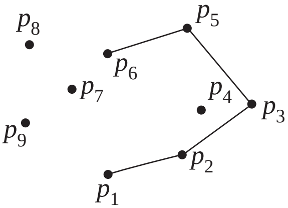
Метод обхода Грэхема



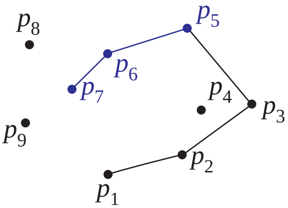
Метод обхода Грэхема



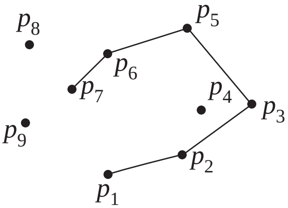
Метод обхода Грэхема



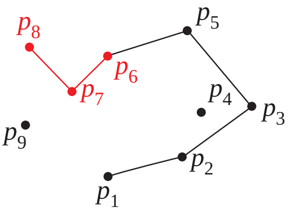
Метод обхода Грэхема



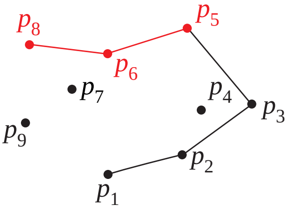
Метод обхода Грэхема



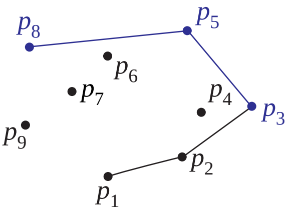
Метод обхода Грэхема



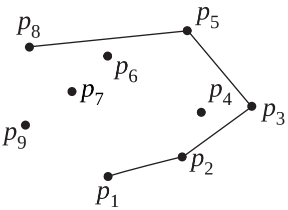
Метод обхода Грэхема



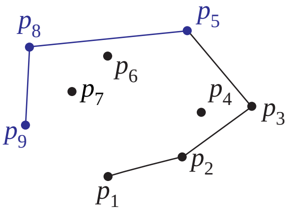
Метод обхода Грэхема



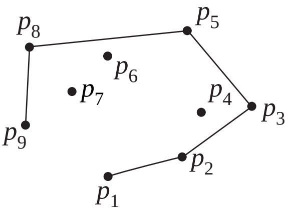
Метод обхода Грэхема



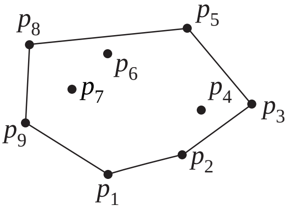
Метод обхода Грэхема



Метод обхода Грэхема



Метод обхода Грэхема



Метод обхода Грэхема

- Вспомогательная структура данных: стек

Метод обхода Грэхема

- Вспомогательная структура данных: стек
- Корректность алгоритма:

Метод обхода Грэхема

- Вспомогательная структура данных: стек
- Корректность алгоритма:
  - Пусть $P_i = \{p_1, \ldots, p_i\}$

Метод обхода Грэхема

- Вспомогательная структура данных: стек
- Корректность алгоритма:
  - Пусть $P_i = \{p_1, \ldots, p_i\}$
  - По завершении i-й итерации стек содержит вершины $\mathcal{CH}(P_i)$

Метод обхода Грэхема

- Вспомогательная структура данных: стек
- Корректность алгоритма:
  - Пусть $P_i = \{p_1, \ldots, p_i\}$
  - По завершении i-й итерации стек содержит вершины $\mathcal{CH}(P_i)$
- Временна́я сложность: $O(n \log n)$

Метод обхода Грэхема

- Вспомогательная структура данных: стек
- Корректность алгоритма:
  - Пусть $P_i = \{p_1, \dots, p_i\}$
  - По завершении i-й итерации стек содержит вершины $\mathcal{CH}(P_i)$
- Временна́я сложность: $O(n \log n)$
  - Нахождение p_1: $O(n)$

Метод обхода Грэхема

- Вспомогательная структура данных: стек
- Корректность алгоритма:
    - Пусть $P_i = \{p_1, \ldots, p_i\}$
    - По завершении i-й итерации стек содержит вершины $\mathcal{CH}(P_i)$
- Временнáя сложность: $O(n \log n)$
    - Нахождение p_1: $O(n)$
    - Сортировка вершин: $O(n \log n)$

Метод обхода Грэхема

- Вспомогательная структура данных: стек
- Корректность алгоритма:
  - Пусть $P_i = \{p_1, \dots, p_i\}$
  - По завершении i-й итерации стек содержит вершины $\mathcal{CH}(P_i)$
- Временнáя сложность: $O(n \log n)$
  - Нахождение p_1: $O(n)$
  - Сортировка вершин: $O(n \log n)$
  - Обход: $O(n)$

Метод обхода Грэхема

- Вспомогательная структура данных: стек
- Корректность алгоритма:
  - Пусть $P_i = \{p_1, \ldots, p_i\}$
  - По завершении i-й итерации стек содержит вершины $\mathcal{CH}(P_i)$
- Временна́я сложность: $O(n \log n)$
  - Нахождение p_1: $O(n)$
  - Сортировка вершин: $O(n \log n)$
  - Обход: $O(n)$
    - Время пропорционально общему числу операций со стеком

Метод обхода Грэхема

- Вспомогательная структура данных: стек
- Корректность алгоритма:
  - Пусть $P_i = \{p_1, \dots, p_i\}$
  - По завершении i-й итерации стек содержит вершины $\mathcal{CH}(P_i)$
- Временна́я сложность: $O(n \log n)$
  - Нахождение p_1: $O(n)$
  - Сортировка вершин: $O(n \log n)$
  - Обход: $O(n)$
    - Время пропорционально общему числу операций со стеком
    - Каждая из вершин $p_2, \dots, p_n$ помещается в стек ровно один раз

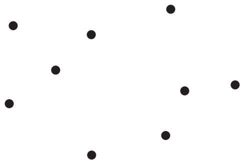
Метод обхода Грэхема

- Вспомогательная структура данных: стек
- Корректность алгоритма:
  - Пусть $P_i = \{p_1, \ldots, p_i\}$
  - По завершении i-й итерации стек содержит вершины $\mathcal{CH}(P_i)$
- Временна́я сложность: $O(n \log n)$
  - Нахождение p_1: $O(n)$
  - Сортировка вершин: $O(n \log n)$
  - Обход: $O(n)$
    - Время пропорционально общему числу операций со стеком
    - Каждая из вершин $p_2, \ldots, p_n$ помещается в стек ровно один раз
    - Следовательно, общее число операций удаления не превосходит $n - 1$

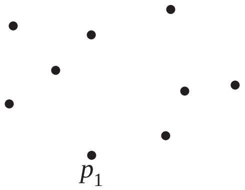
Метод обхода Грэхема

- Вспомогательная структура данных: стек
- Корректность алгоритма:
  - Пусть $P_i = \{p_1, \ldots, p_i\}$
  - По завершении i-й итерации стек содержит вершины $\mathcal{CH}(P_i)$
- Временнáя сложность: $O(n \log n)$
  - Нахождение p_1: $O(n)$
  - Сортировка вершин: $O(n \log n)$
  - Обход: $O(n)$
    - Время пропорционально общему числу операций со стеком
    - Каждая из вершин p_2, . . . , p_n помещается в стек ровно один раз
    - Следовательно, общее число операций удаления не превосходит $n - 1$
- Затраты памяти: $O(n)$

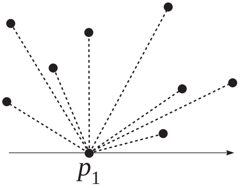
Метод Джарвиса



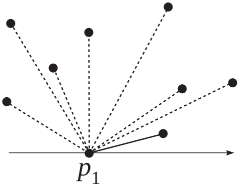
Метод Джарвиса



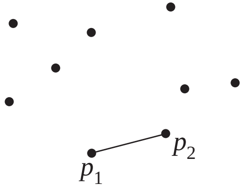
Метод Джарвиса



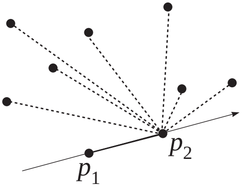
Метод Джарвиса



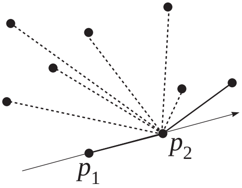
Метод Джарвиса



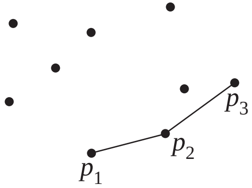
Метод Джарвиса



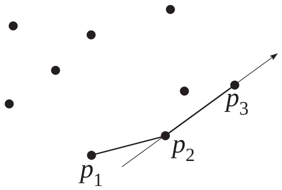
Метод Джарвиса



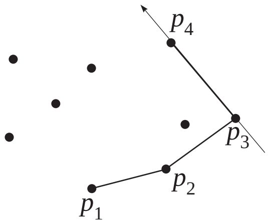
Метод Джарвиса



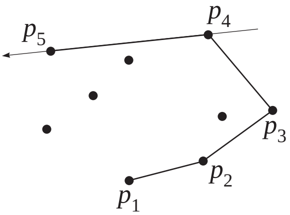
Метод Джарвиса



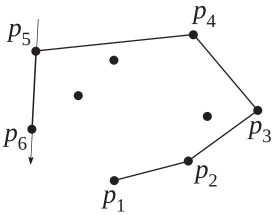
Метод Джарвиса



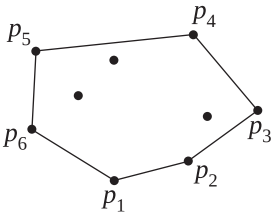
Метод Джарвиса



Метод Джарвиса



Метод Джарвиса



Метод Джарвиса

- Корректность:

Метод Джарвиса

- Корректность:
  - На каждом шаге строится очередное ребро $\mathcal{CH}(P)$

Метод Джарвиса

- Корректность:
  - На каждом шаге строится очередное ребро $\mathcal{CH}(P)$
- Временна́я сложность: $O(nh)$, где h – число вершин $\mathcal{CH}(P)$

Метод Джарвиса

- Корректность:
  - На каждом шаге строится очередное ребро $\mathcal{CH}(P)$
- Временна́я сложность: $O(nh)$, где h – число вершин $\mathcal{CH}(P)$
  - Алгоритм, *чувствительный к выходу*

Метод Джарвиса

- Корректность:
  - На каждом шаге строится очередное ребро $\mathcal{CH}(P)$
- Временна́я сложность: $O(nh)$, где h – число вершин $\mathcal{CH}(P)$
  - Алгоритм, *чувствительный к выходу*
- Затраты памяти: $O(n)$

Хронология

- 1972: метод обхода Грэхема, $O(n \log n)$

Хронология

- 1972: метод обхода Грэхема, $O(n \log n)$
- 1973: метод Джарвиса, $O(nh)$

Хронология

- 1972: метод обхода Грэхема, $O(n \log n)$
- 1973: метод Джарвиса, $O(nh)$
- 1986: алгоритм Киркпатрика и Зейделя, $O(n \log h)$

Хронология

- 1972: метод обхода Грэхема, $O(n \log n)$
- 1973: метод Джарвиса, $O(nh)$
- 1986: алгоритм Киркпатрика и Зейделя, $O(n \log h)$
  - Сложный алгоритм

Хронология

- 1972: метод обхода Грэхема, $O(n \log n)$
- 1973: метод Джарвиса, $O(nh)$
- 1986: алгоритм Киркпатрика и Зейделя, $O(n \log h)$
  - Сложный алгоритм
- 1996: алгоритм Чена, $O(n \log h)$
  - Простой алгоритм

Алгоритм Чена

- Зафиксируем $m \leq n$

Алгоритм Чена

- Зафиксируем $m \leq n$
- Положим $r = \left\lceil \frac{n}{m} \right\rceil$

Алгоритм Чена

- Зафиксируем $m \leq n$
- Положим $r = \lceil \frac{n}{m} \rceil$
- Разобьем P на r подмножеств $P_1, \ldots, P_r$
  - $|P_1| = \cdots = |P_{r-1}| = m$, $|P_r| \leq m$

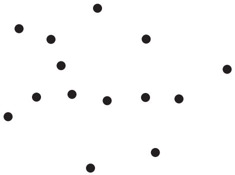
Алгоритм Чена

- Зафиксируем $m \leq n$
- Положим $r = \lceil \frac{n}{m} \rceil$
- Разобьем P на r подмножеств $P_1, \ldots, P_r$
  - $|P_1| = \cdots = |P_{r-1}| = m$, $|P_r| \leq m$
- Для каждого подмножества P_i построим $\mathcal{CH}(P_i)$ методом обхода Грэхема

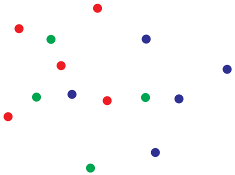
Алгоритм Чена

- Зафиксируем $m \leq n$
- Положим $r = \lceil \frac{n}{m} \rceil$
- Разобьем P на r подмножеств $P_1, \ldots, P_r$
  - $|P_1| = \cdots = |P_{r-1}| = m$, $|P_r| \leq m$
- Для каждого подмножества P_i построим $\mathcal{CH}(P_i)$ методом обхода Грэхема
- К полученным выпуклым оболочкам применим метод Джарвиса

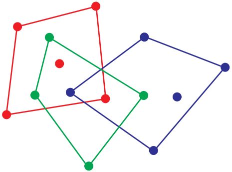
Алгоритм Чена



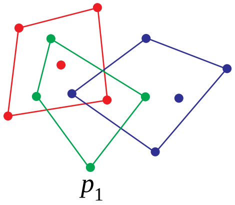
Алгоритм Чена



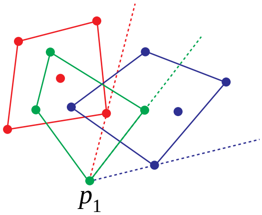
Алгоритм Чена



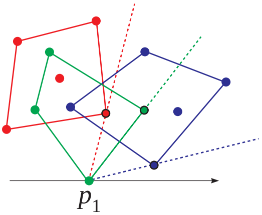
Алгоритм Чена



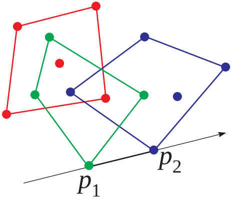
Алгоритм Чена



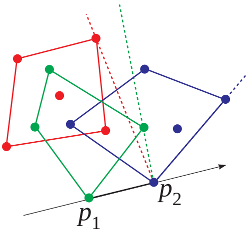
Алгоритм Чена



Алгоритм Чена



Алгоритм Чена



Предварительный анализ сложности

- Предобработка:

Предварительный анализ сложности

- Предобработка:
  - Построение $\mathcal{CH}(P_i)$: $O(m \log m)$ времени

Предварительный анализ сложности

- Предобработка:
  - Построение $\mathcal{CH}(P_i)$: $O(m \log m)$ времени
  - Построение всех выпуклых оболочек: $O(rm \log m) = O(n \log m)$ времени

Предварительный анализ сложности

- Предобработка:
  - Построение $\mathcal{CH}(P_i)$: $O(m \log m)$ времени
  - Построение всех выпуклых оболочек: $O(rm \log m) = O(n \log m)$ времени
- Основной цикл:

Предварительный анализ сложности

- Предобработка:
  - Построение $\mathcal{CH}(P_i)$: $O(m \log m)$ времени
  - Построение всех выпуклых оболочек: $O(rm \log m) = O(n \log m)$ времени
- Основной цикл:
  - Построение опорной прямой к $\mathcal{CH}(P_i)$: $O(\log m)$ времени

Предварительный анализ сложности

- Предобработка:
  - Построение $\mathcal{CH}(P_i)$: $O(m \log m)$ времени
  - Построение всех выпуклых оболочек: $O(rm \log m) = O(n \log m)$ времени
- Основной цикл:
  - Построение опорной прямой к $\mathcal{CH}(P_i)$: $O(\log m)$ времени
  - Нахождение очередной вершины $\mathcal{CH}(P)$: $O(r \log m)$ времени

Предварительный анализ сложности

- Предобработка:
  - Построение $\mathcal{CH}(P_i)$: $O(m \log m)$ времени
  - Построение всех выпуклых оболочек: $O(rm \log m) = O(n \log m)$ времени
- Основной цикл:
  - Построение опорной прямой к $\mathcal{CH}(P_i)$: $O(\log m)$ времени
  - Нахождение очередной вершины $\mathcal{CH}(P)$: $O(r \log m)$ времени
  - Нахождение всех вершин $\mathcal{CH}(P)$: $O(hr \log m) = O(h \cdot \frac{n}{m} \cdot \log m)$ времени

Предварительный анализ сложности

- Предобработка:
  - Построение $\mathcal{CH}(P_i)$: $O(m \log m)$ времени
  - Построение всех выпуклых оболочек: $O(rm \log m) = O(n \log m)$ времени
- Основной цикл:
  - Построение опорной прямой к $\mathcal{CH}(P_i)$: $O(\log m)$ времени
  - Нахождение очередной вершины $\mathcal{CH}(P)$: $O(r \log m)$ времени
  - Нахождение всех вершин $\mathcal{CH}(P)$: $O(hr \log m) = O(h \cdot \frac{n}{m} \cdot \log m)$ времени
- Общее время работы алгоритма: $O\left(\left(n + h \cdot \frac{n}{m}\right) \log m\right)$

Предварительный анализ сложности

- Предобработка:
  - Построение $\mathcal{CH}(P_i)$: $O(m \log m)$ времени
  - Построение всех выпуклых оболочек: $O(rm \log m) = O(n \log m)$ времени
- Основной цикл:
  - Построение опорной прямой к $\mathcal{CH}(P_i)$: $O(\log m)$ времени
  - Нахождение очередной вершины $\mathcal{CH}(P)$: $O(r \log m)$ времени
  - Нахождение всех вершин $\mathcal{CH}(P)$: $O(hr \log m) = O(h \cdot \frac{n}{m} \cdot \log m)$ времени
- Общее время работы алгоритма: $O\left(\left(n + h \cdot \frac{n}{m}\right) \log m\right)$
- При $m = h$: $O(n \log h)$

Частичное построение выпуклой оболочки

PartialHull(P, m)

$r \leftarrow \lceil \frac{n}{m} \rceil$
$P \rightarrow P_1, \ldots, P_r$
for $i \leftarrow 1$ **to** r
    do Построить $\mathcal{CH}(P_i)$
$p_1 \leftarrow$ точка из P с наименьшей ординатой
(в случае неоднозначности – самая левая из таковых)
for $k \leftarrow 1$ **to** m
    do Найти вершину p_{k+1} выпуклой оболочки $\mathcal{CH}(P)$
        if $p_{k+1} = p_1$
            then return (“$(p_1, \ldots, p_k)$”)
return “m слишком мало”

Частичное построение выпуклой оболочки

$\textsc{PartialHull}(P, m)$

$r \leftarrow \lceil \frac{n}{m} \rceil$
$P \rightarrow P_1, \ldots, P_r$
for $i \leftarrow 1$ **to** r $\triangleright$ $O(r \cdot m \log m) = O(n \log m)$
  do Построить $\mathcal{CH}(P_i)$
$p_1 \leftarrow$ точка из P с наименьшей ординатой
(в случае неоднозначности – самая левая из таковых)
for $k \leftarrow 1$ **to** m $\triangleright$ $O(m \cdot r \log m) = O(n \log m)$
  do Найти вершину p_{k+1} выпуклой оболочки $\mathcal{CH}(P)$
    if $p_{k+1} = p_1$
      then return (“$(p_1, \ldots, p_k)$”)
return “m слишком мало”

Частичное построение выпуклой оболочки

$\textsc{PartialHull}(P, m)$

$r \leftarrow \lceil \frac{n}{m} \rceil$
$P \rightarrow P_1, \ldots, P_r$
for $i \leftarrow 1$ **to** r
  do Построить $\mathcal{CH}(P_i)$
$p_1 \leftarrow$ точка из P с наименьшей ординатой
(в случае неоднозначности – самая левая из таковых)
for $k \leftarrow 1$ **to** m
  do Найти вершину p_{k+1} выпуклой оболочки $\mathcal{CH}(P)$
    if $p_{k+1} = p_1$
      then return ("$(p_1, \ldots, p_k)$")
return "m слишком мало"

Время работы: $O(n \log m)$

Итоговый алгоритм

$\textsc{ConvexHull}(P)$

for $t \leftarrow 1, 2, 3, \ldots$
    do $m \leftarrow \min\{2^{2^t}, n\}$
        $L \leftarrow \textsc{PartialHull}(P, m)$
        if $L \neq$ "m слишком мало"
            then return L

Итоговый алгоритм

ConvexHull(P)

for $t \leftarrow 1, 2, 3, \ldots$
    do $m \leftarrow \min\{2^{2^t}, n\}$
        $L \leftarrow$ PartialHull(P,m)
        if $L \neq$ "m слишком мало"
            then return L

Алгоритм завершит работу при $t = \lceil \log \log h \rceil$

Анализ сложности

- Время выполнения итерации t: $O(n \log 2^{2^t}) = O(n \cdot 2^t)$

Анализ сложности

- Время выполнения итерации t: $O(n \log 2^{2^t}) = O(n \cdot 2^t)$
- Общее время работы алгоритма: $\sum_{t=1}^{\lceil \log \log h \rceil} O(n \cdot 2^t)$

Анализ сложности

- Время выполнения итерации t: $O(n \log 2^{2^t}) = O(n \cdot 2^t)$
- Общее время работы алгоритма: $\sum_{t=1}^{\lceil \log \log h \rceil} O(n \cdot 2^t)$
- $\sum_{t=1}^{\lceil \log \log h \rceil} n \cdot 2^t < n \cdot 2^{\lceil \log \log h \rceil + 1} < 4n \cdot 2^{\log \log h} = 4n \log h$

Анализ сложности

- Время выполнения итерации t: $O(n \log 2^{2^t}) = O(n \cdot 2^t)$
- Общее время работы алгоритма: $\sum_{t=1}^{\lceil \log \log h \rceil} O(n \cdot 2^t)$
- $\sum_{t=1}^{\lceil \log \log h \rceil} n \cdot 2^t < n \cdot 2^{\lceil \log \log h \rceil + 1} < 4n \cdot 2^{\log \log h} = 4n \log h$
- Временнáя сложность алгоритма Чена: $O(n \log h)$

Анализ сложности

- Время выполнения итерации t: $O(n \log 2^{2^t}) = O(n \cdot 2^t)$
- Общее время работы алгоритма: $\sum_{t=1}^{\lceil \log \log h \rceil} O(n \cdot 2^t)$
- $\sum_{t=1}^{\lceil \log \log h \rceil} n \cdot 2^t < n \cdot 2^{\lceil \log \log h \rceil + 1} < 4n \cdot 2^{\log \log h} = 4n \log h$
- Временнáя сложность алгоритма Чена: $O(n \log h)$
- Затраты памяти: $O(n)$

План доклада

1. Выпуклые оболочки
2. Минимальная описанная окружность
3. Треугольник минимальной площади

Минимальная описанная окружность

- Дано множество P из n точек на плоскости

Минимальная описанная окружность

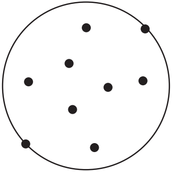
- Дано множество P из n точек на плоскости
- Построить окружность минимального радиуса, содержащую внутри все точки из P

Свойства

- Минимальная описанная окружность для P
    - определена однозначно

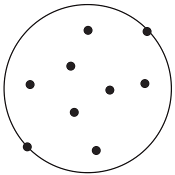
Свойства

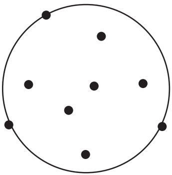
- Минимальная описанная окружность для P
  - определена однозначно
  - проходит
    - либо ровно через две точки из P, которые определяют ее диаметр,



Свойства

- Минимальная описанная окружность для P
  - определена однозначно
  - проходит
    - либо ровно через две точки из P, которые определяют ее диаметр,
    - либо по крайней мере через три точки из P





Обозначения

- Пусть $1 \leq i \leq n$

Обозначения

- Пусть $1 \leq i \leq n$
- $P_i = \{p_1, \ldots, p_i\}$

Обозначения

- Пусть $1 \leq i \leq n$
- $P_i = \{p_1, \ldots, p_i\}$
- C_i – минимальная описанная окружность для P_i

Генерация случайной перестановки

$\textsc{RandomPermutation}(A)$

for $k \leftarrow n$ **downto** 2
    do $rndIndex \leftarrow \textsc{Random}(k)$
        $A[k] \leftrightarrow A[rndIndex]$

- A – множество элементов, представленное при помощи массива
- $\textsc{Random}(k)$ генерирует случайное целое из диапазона $[1, k]$ за время $O(1)$

Генерация случайной перестановки

```
RandomPermutation(A)
for k ← n downto 2
    do rndIndex ← Random(k)
        A[k] ↔ A[rndIndex]
```

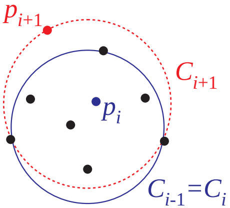
- A – множество элементов, представленное при помощи массива
- $\textsc{Random}(k)$ генерирует случайное целое из диапазона $[1, k]$ за время $O(1)$
- $\forall a \in A$, вероятность нахождения a на месте j в случайной перестановке составляет $1/n$, где $1 \leq j \leq n$

Лемма 1

- Если p_i – внутри C_{i-1}, то $C_i = C_{i-1}$
- Если p_i – вне C_{i-1}, то C_i проходит через p_i

Лемма 1

- Если p_i – внутри C_{i-1}, то $C_i = C_{i-1}$
- Если p_i – вне C_{i-1}, то C_i проходит через p_i



Построение

$\textsc{MinCircle}(P)$

Сгенерировать случайную перестановку точек из P
$C_2 \leftarrow$ окружность с диаметром $p_1 p_2$
for $i \leftarrow 3$ **to** n
  do if p_i – внутри C_{i-1}
    then $C_i \leftarrow C_{i-1}$
    else $C_i \leftarrow \textsc{MinCircle-1}(P_{i-1},\ p_i)$

Лемма 2

- q – точка на плоскости, такая, что существует описанная окружность для P, проходящая через q

Лемма 2

- q – точка на плоскости, такая, что существует описанная окружность для P, проходящая через q
- C_i^q – минимальная описанная окружность для P_i, проходящая через q

Лемма 2

- q – точка на плоскости, такая, что существует описанная окружность для P, проходящая через q
- C_i^q – минимальная описанная окружность для P_i, проходящая через q
- Если p_i – внутри C_{i-1}^q, то $C_i^q = C_{i-1}^q$
- Если p_i – вне C_{i-1}^q, то C_i^q проходит через p_i

Построение: одна фиксированная точка

$\textsc{MinCircle-1}(P, q)$

Сгенерировать случайную перестановку точек из P
$C_1 \leftarrow$ окружность с диаметром $p_1 q$
for $j \leftarrow 2$ **to** n
    do if p_j – внутри C_{j-1}
        then $C_j \leftarrow C_{j-1}$
        else $C_j \leftarrow \textsc{MinCircle-2}(P_{j-1},\ p_j,\ q)$

Лемма 3

- q_1, q_2 – точки на плоскости, такие, что существует описанная окружность для P, проходящая через q_1 и q_2

Лемма 3

- q_1, q_2 – точки на плоскости, такие, что существует описанная окружность для P, проходящая через q_1 и q_2
- $C_i^{q_1,q_2}$ – минимальная описанная окружность для P_i, проходящая через q_1 и q_2

Лемма 3

- q_1, q_2 – точки на плоскости, такие, что существует описанная окружность для P, проходящая через q_1 и q_2
- $C_i^{q_1,q_2}$ – минимальная описанная окружность для P_i, проходящая через q_1 и q_2
- Если p_i – внутри $C_{i-1}^{q_1,q_2}$, то $C_i^{q_1,q_2} = C_{i-1}^{q_1,q_2}$
- Если p_i – вне $C_{i-1}^{q_1,q_2}$, то $C_i^{q_1,q_2}$ проходит через p_i

Построение: две фиксированных точки

$\textsc{MinCircle-2}(P, q_1, q_2)$

$C_0 \leftarrow$ окружность с диаметром $q_1 q_2$
for $k \leftarrow 1$ **to** n
    do if p_k – внутри C_{k-1}
        then $C_k \leftarrow C_{k-1}$
        else $C_k \leftarrow$ окружность, проходящая через q_1, q_2 и p_k

Анализ сложности

- MinCircle-2(P):

Анализ сложности

- MinCircle-2(P):
  - Время работы: $O(n)$

Анализ сложности

- MinCircle-2(P):
  - Время работы: $O(n)$
- MinCircle-1(P):

Анализ сложности

- $\textsc{MinCircle-2}(P)$:
  - Время работы: $O(n)$
- $\textsc{MinCircle-1}(P)$:
  - Вероятность вызова $\textsc{MinCircle-2}$ на итерации j не превосходит $2/j$

Анализ сложности

- $\textsc{MinCircle-2}(P)$:
  - Время работы: $O(n)$
- $\textsc{MinCircle-1}(P)$:
  - Вероятность вызова $\textsc{MinCircle-2}$ на итерации j не превосходит $2/j$
  - Ожидаемое время работы: $O(n) + \sum\limits_{j=2}^{n} O(j)\frac{2}{j} = O(n)$

Анализ сложности

- $\textsc{MinCircle-2}(P)$:
  - Время работы: $O(n)$
- $\textsc{MinCircle-1}(P)$:
  - Вероятность вызова $\textsc{MinCircle-2}$ на итерации j не превосходит $2/j$
  - Ожидаемое время работы: $O(n) + \sum_{j=2}^{n} O(j)\frac{2}{j} = O(n)$
- $\textsc{MinCircle}(P)$:

Анализ сложности

- $\textsc{MinCircle-2}(P)$:
  - Время работы: $O(n)$
- $\textsc{MinCircle-1}(P)$:
  - Вероятность вызова $\textsc{MinCircle-2}$ на итерации j не превосходит $2/j$
  - Ожидаемое время работы: $O(n) + \sum\limits_{j=2}^{n} O(j)\frac{2}{j} = O(n)$
- $\textsc{MinCircle}(P)$:
  - Вероятность вызова $\textsc{MinCircle-1}$ на итерации i не превосходит $3/i$

Анализ сложности

- $\textsc{MinCircle-2}(P)$:
  - Время работы: $O(n)$
- $\textsc{MinCircle-1}(P)$:
  - Вероятность вызова $\textsc{MinCircle-2}$ на итерации j не превосходит $2/j$
  - Ожидаемое время работы: $O(n) + \sum_{j=2}^{n} O(j)\frac{2}{j} = O(n)$
- $\textsc{MinCircle}(P)$:
  - Вероятность вызова $\textsc{MinCircle-1}$ на итерации i не превосходит $3/i$
  - Ожидаемое время работы: $O(n) + \sum_{i=3}^{n} O(i)\frac{3}{i} = O(n)$

Анализ сложности

- $\textsc{MinCircle-2}(P)$:
  - Время работы: $O(n)$
- $\textsc{MinCircle-1}(P)$:
  - Вероятность вызова $\textsc{MinCircle-2}$ на итерации j не превосходит $2/j$
  - Ожидаемое время работы: $O(n) + \sum_{j=2}^{n} O(j)\frac{2}{j} = O(n)$
- $\textsc{MinCircle}(P)$:
  - Вероятность вызова $\textsc{MinCircle-1}$ на итерации i не превосходит $3/i$
  - Ожидаемое время работы: $O(n) + \sum_{i=3}^{n} O(i)\frac{3}{i} = O(n)$
- Ожидаемое время построения минимальной описанной окружности: $O(n)$

Анализ сложности

- $\textsc{MinCircle-2}(P)$:
  - Время работы: $O(n)$
- $\textsc{MinCircle-1}(P)$:
  - Вероятность вызова $\textsc{MinCircle-2}$ на итерации j не превосходит $2/j$
  - Ожидаемое время работы: $O(n) + \sum_{j=2}^{n} O(j)\frac{2}{j} = O(n)$
- $\textsc{MinCircle}(P)$:
  - Вероятность вызова $\textsc{MinCircle-1}$ на итерации i не превосходит $3/i$
  - Ожидаемое время работы: $O(n) + \sum_{i=3}^{n} O(i)\frac{3}{i} = O(n)$
- Ожидаемое время построения минимальной описанной окружности: $O(n)$
- Затраты памяти: $O(n)$

План доклада

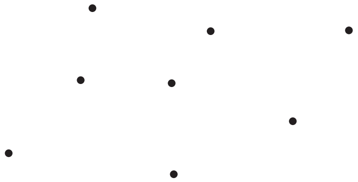
Треугольник минимальной площади

- Дано множество P из n точек на плоскости

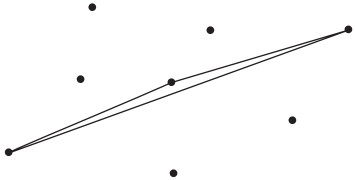
Треугольник минимальной площади

- Дано множество P из n точек на плоскости
- Найти треугольник минимальной площади с вершинами в трех различных вершинах из P

Нелокальность

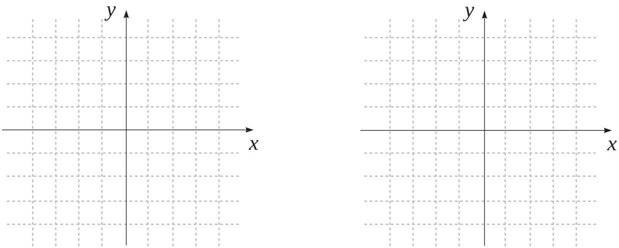


Нелокальность



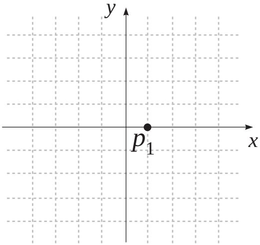
Преобразование двойственности

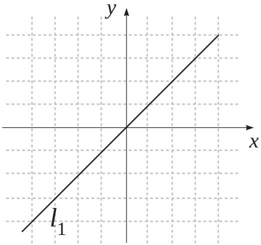
$$p = (a, b) \quad \rightarrow \quad \ell\colon\ y = ax + b$$
$$\ell\colon\ y = kx + d \quad \rightarrow \quad p = (-k, d)$$



Преобразование двойственности

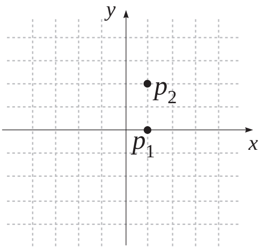
$$p = (a, b) \quad \to \quad \ell\colon\ y = ax + b$$
$$\ell\colon\ y = kx + d \quad \to \quad p = (-k, d)$$

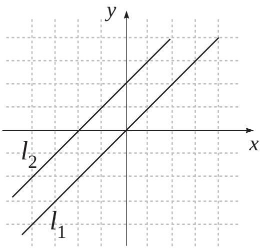




Преобразование двойственности

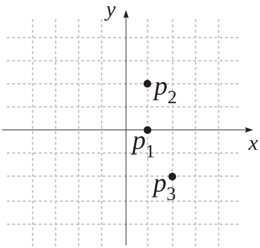
$$p = (a, b) \quad \rightarrow \quad \ell\colon\ y = ax + b$$
$$\ell\colon\ y = kx + d \quad \rightarrow \quad p = (-k, d)$$

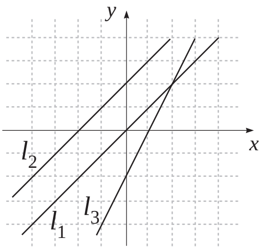




Преобразование двойственности

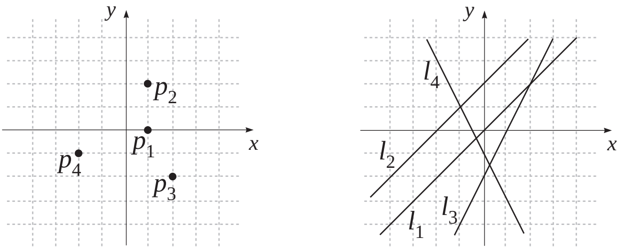
$$p = (a, b) \quad \rightarrow \quad \ell\colon\ y = ax + b$$
$$\ell\colon\ y = kx + d \quad \rightarrow \quad p = (-k, d)$$





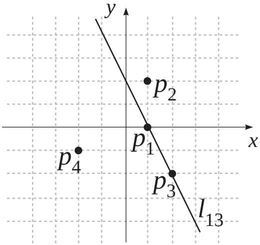
Преобразование двойственности

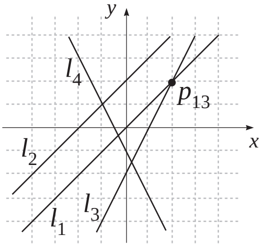
$$p = (a, b) \quad \to \quad \ell\colon\ y = ax + b$$
$$\ell\colon\ y = kx + d \quad \to \quad p = (-k, d)$$



Преобразование двойственности

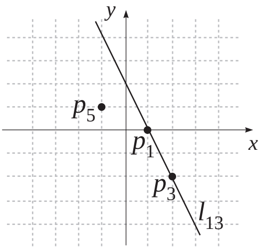
$$p = (a, b) \quad \to \quad \ell\colon\ y = ax + b$$
$$\ell\colon\ y = kx + d \quad \to \quad p = (-k, d)$$

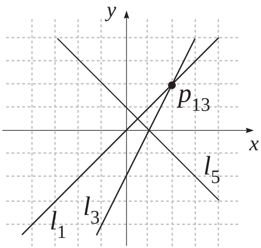




Преобразование двойственности

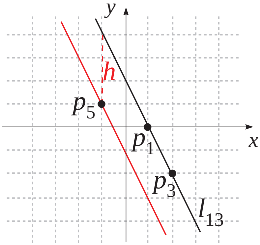
$$p = (a, b) \quad \rightarrow \quad \ell\colon\ y = ax + b$$
$$\ell\colon\ y = kx + d \quad \rightarrow \quad p = (-k, d)$$

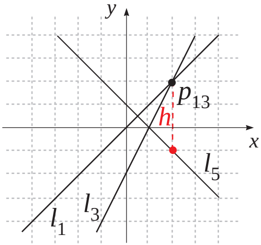




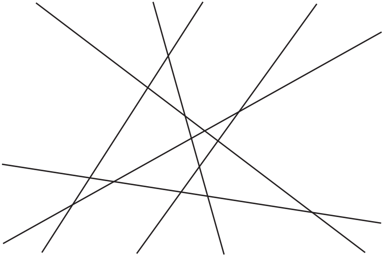
Преобразование двойственности

$$p = (a, b) \quad \rightarrow \quad \ell\colon\ y = ax + b$$
$$\ell\colon\ y = kx + d \quad \rightarrow \quad p = (-k, d)$$

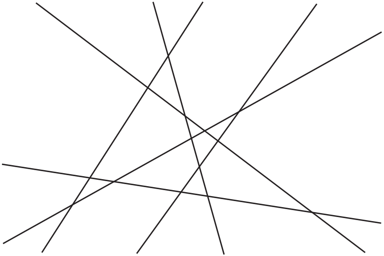




Двойственная постановка задачи

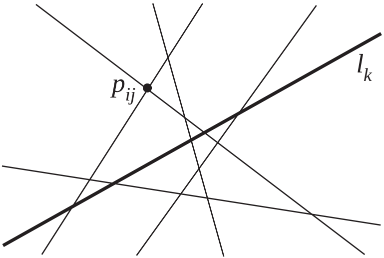


Двойственная постановка задачи



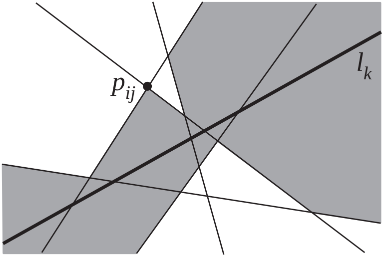
- Пусть $\triangle p_i p_j p_k$ – треугольник минимальной площади

Двойственная постановка задачи



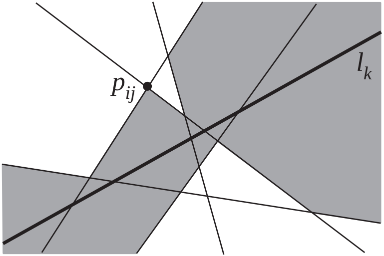
- Пусть $\triangle p_i p_j p_k$ – треугольник минимальной площади
- На двойственной плоскости: p_{ij} – вершина грани, инцидентной l_k

Двойственная постановка задачи



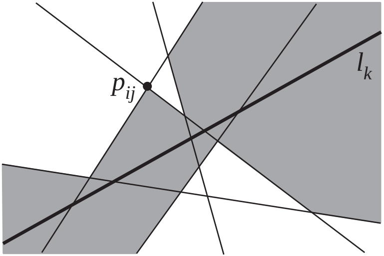
- Пусть $\triangle p_i p_j p_k$ – треугольник минимальной площади
- На двойственной плоскости: p_{ij} – вершина грани, инцидентной l_k
- Для каждой прямой необходимо просмотреть $O(n)$ вершин

Двойственная постановка задачи



- Пусть $\triangle p_i p_j p_k$ – треугольник минимальной площади
- На двойственной плоскости: p_{ij} – вершина грани, инцидентной l_k
- Для каждой прямой необходимо просмотреть $O(n)$ вершин
- Временна́я сложность: $O(n^2)$

Двойственная постановка задачи



- Пусть $\triangle p_i p_j p_k$ – треугольник минимальной площади
- На двойственной плоскости: p_{ij} – вершина грани, инцидентной l_k
- Для каждой прямой необходимо просмотреть $O(n)$ вершин
- Временна́я сложность: $O(n^2)$
- Затраты памяти: $O(n^2)$

Литература: монографии и курс лекций

1. M. de Berg, O. Cheong, M. van Kreveld, M. Overmars, "Computational Geometry: Algorithms and Applications", Third Edition, Springer, 2008.
2. J. O'Rourke, "Computational Geometry in C", Second Edition, Cambridge University Press, 1998.
3. J.-D. Boissonnat, M. Yvinec, "Géométrie algorithmique", Ediscience international, Paris, 1995.
4. Ф. Препарата, М. Шеймос, "Вычислительная геометрия: введение", пер. с англ., М., Мир, 1989.
5. D. Mount, "Computational Geometry: Lecture Notes", Fall 2002, `http://www.cs.umd.edu/~mount/754/Lects/754lects.pdf`

Литература: статьи

1. R. L. Graham, "An efficient algorithms for determining the convex hull of a finite planar set", Info. Proc. Lett., 1:132–133, 1972.
2. R. A. Jarvis, "On the identification of the convex hull of a finite set of points in the plane", Info. Proc. Lett., 2:18–21, 1973.
3. D. G. Kirkpatrick, R. Seidel, "The ultimate planar convex hull algorithm", SIAM J. Comput., 15(1):287–299, 1986.
4. T. Chan, "Optimal output-sensitive convex hull algorithms in two and three dimensions", Discr. Comp. Geom., 16:361–368, 1996.
5. E. Welzl, "Smallest enclosing disks (balls and ellipsoids)", in H. Maurer (Ed.), "New Results and New Trends in Computer Science", LNCS, 555:359–370, Springer, 1991.
6. B. Chazelle, L. J. Guibas, D. T. Lee, "The power of geometric duality", BIT, 25:76–90, 1985.