Вычислительная геометрия

Борис Золотов   Матвей Магин

14 июня 2022 г.

Летняя школа МКН СПбГУ

Содержание

Algorithms to get a feeling

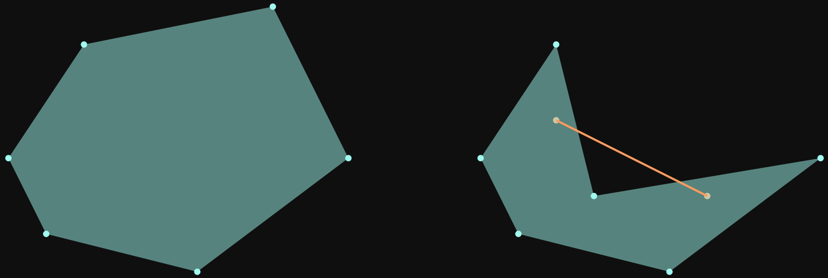
- Выпуклая оболочка
- Триангуляции многоугольников
- Задача о картинной галерее

Algorithms to get a feeling

Выпуклое множество

Определение

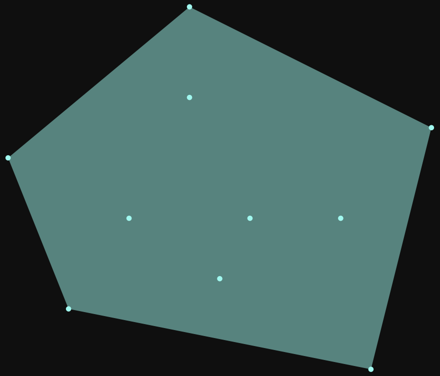
Множество S — выпуклое, если оно вместе с любыми двумя точками содержит отрезок между ними.



Выпуклая оболочка

Определение

Выпуклая оболочка $\mathcal{CH}(S)$ множества S — наименьшее выпуклое множество, содержащее S.



Вычисление выпуклой оболочки

Задача

Дано множество $S \subset \mathbb{R}^2$, $|S| = n$. Требуется найти координаты вершин его выпуклой оболочки $\mathcal{CH}(S)$.

Вычисление выпуклой оболочки

Есть много алгоритмов вычисления выпуклой оболочки на плоскости. Большинство из них напоминают алгоритмы сортировок, к примеру

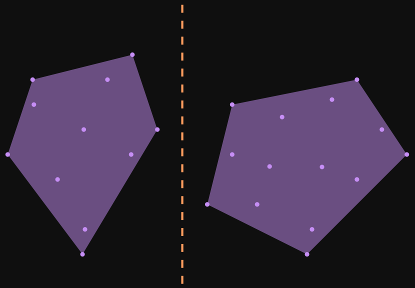
- Алгоритм Джарвиса – Selectiont Sort.
- Quick Hull – Quick Sort.
- Алгоритм «Разделяй и властвуй» – Merge Sort.

Мы рассмотрим алгоритм «Devide-and-Conquer» из них.
Все алгоритмы «Devide-and-Conquer» имеют одну идею:

- Разбить задачу на подзадачи, от них вызываться рекурсивно.
- Научиться быстро сливать подзадачи.

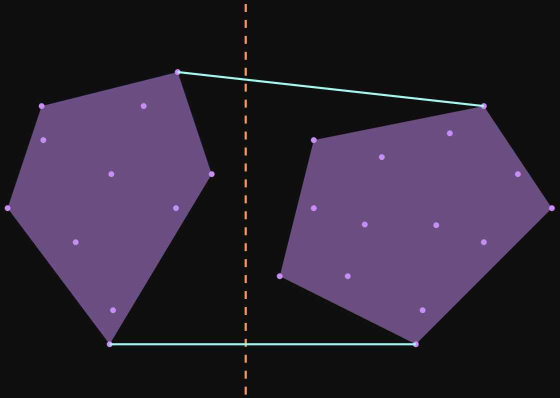
Алгоритм $D\&C$ для $\mathcal{CH}$: описание

- $n \leq 3 \Rightarrow$ «brute force».
- $n \geq 4 \Rightarrow$ разбиваем S на два примерно равных подмножества по x–координате, вызываемся на них рекурсивно.



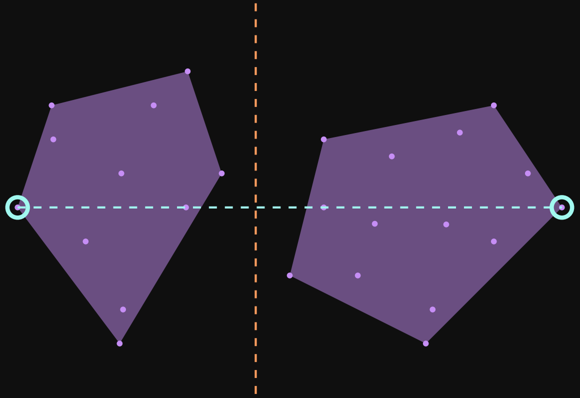
Алгоритм $D\&C$ для $\mathcal{CH}$: слияние подзадач

Для слияния подзадач будем считать верхнюю и нижнюю касательные.



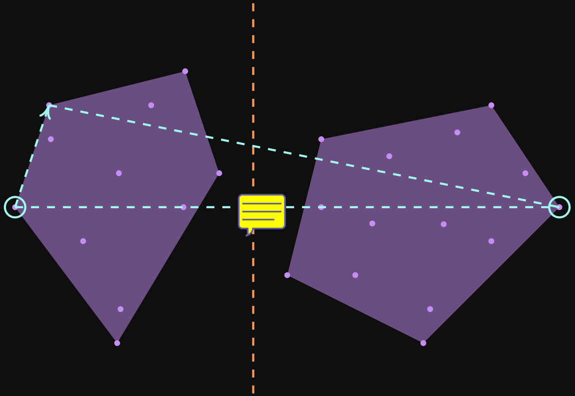
Алгоритм $D\&C$ для $\mathcal{CH}$: верхняя и нижняя касательные

Идея вычисления: поднимаем тот конец,отрезка который можем поднять.



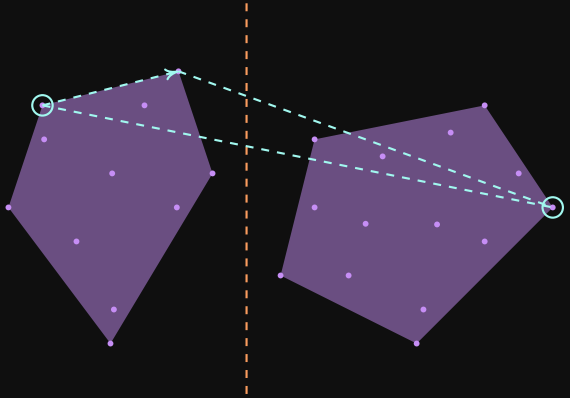
Алгоритм $D\&C$ для $\mathcal{CH}$: верхняя и нижняя касательные

Идея вычисления: поднимаем тот конец отрезка, который можем поднять.



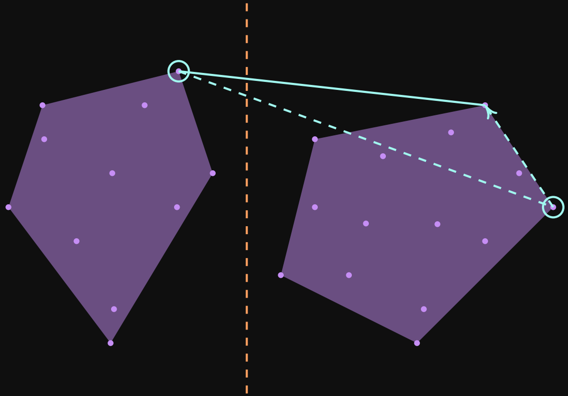
Алгоритм $D\&C$ для $\mathcal{CH}$: верхняя и нижняя касательные

Идея вычисления: поднимаем тот конец отрезка, который можем поднять.



Алгоритм $D\&C$ для $\mathcal{CH}$: верхняя и нижняя касательные

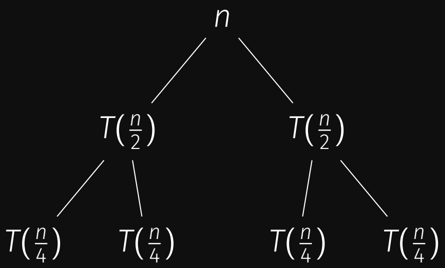
Идея вычисления: поднимаем тот конец отрезка, который можем поднять.



Алгоритм $D\&C$ для $\mathcal{CH}$: оценка времени работы

Количество точек в подзадаче сокращается хотя бы в два раза, на слияние мы тратим линейное время

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$$



Алгоритм $D\&C$ для $\mathcal{CH}$: оценка времени работы

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$$

Распишем это:

$$T(n) = O\left(n \cdot \left(\frac{2}{2} + \left(\frac{2}{2}\right)^2 + \ldots + \left(\frac{2}{2}\right)^{\log_2(n)}\right)\right) = O\left(n \cdot \sum_{k=1}^{\log_2 n} 1\right) = O(n \log_2(n)).$$

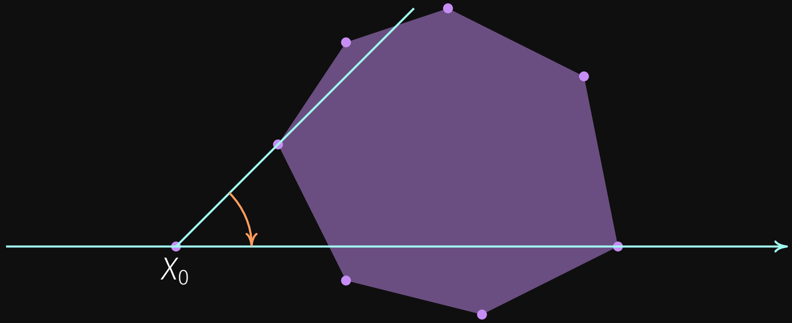
Касательная к выпуклому многоугольнику

Задача

Дан выпуклый многоугольник $\mathcal{P} \subset \mathbb{R}^2$ и точка $X_0(x_0, y_0) \in \mathbb{R}^2$. Найти касательную к $\mathcal{P}$ из точки $X(x_0, y_0)$.

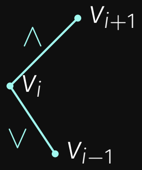
Касательная к выпуклому многоугольнику: алгоритм

Каждой вершине сопоставим *аргумент* – угол, под которым она видна их точки X_0.

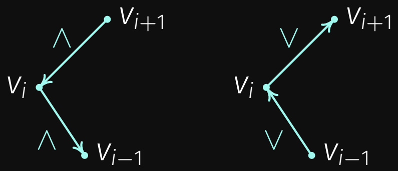


Касательная к выпуклому многоугольнику: алгоритм

У нужной вершины v_i аргументы будут расположены так:

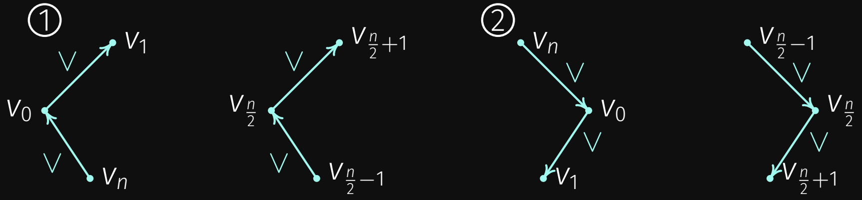


Как понять, с какой стороны нужная вершина:

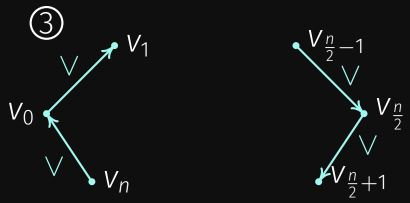


Касательная к выпуклому многоугольнику: алгоритм

- Выберем две вершины на противоположных сторонах – v_0 и $v_{\frac{n}{2}}$, аргументы их и их соседей.
- Получим один из случаев:



Касательная к выпуклому многоугольнику: алгоритм



- Каждый раз отбрасываем половину вершин.
- После отбрасывания половины добавляем вершину посередине нового промежутка и продолжаем.

Касательная к выпуклому многоугольнику: время работы

Каждый следующий запрос – $O(1)$ проверок ориентации. Всего запросов – $O(\log(n))$, а значит, время работы – $O(\log(n))$.

Упражнение.

Придумайте алгоритм выпуклой оболочки, использующий алгоритм построения касательной и алгоритм $D\&C$. Время работы алгоритма должно составлять $O(n\log(h))$, где h – количество вершин в выпуклой оболочке.

Лемма о триангуляции

Лемма (О триангуляции)

Всякий многоугольник можно диагоналями разбить на треугольники, причем полученный граф красится в 3 цвета.

Доказательство.

Индукция по числу вершин.

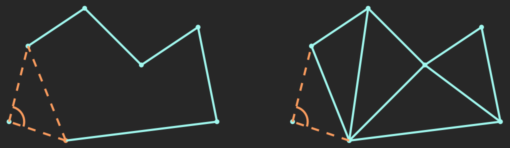
База: $n = 3$.

Переход: Находим вершину с углом $< 180°$.

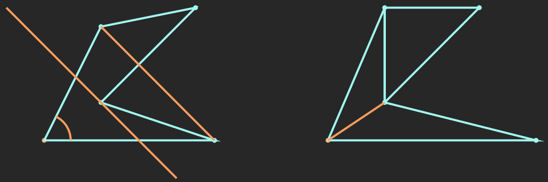
- Отрезок между соседними вершинами лежит в многоугольнике: Отрезаем вершину, красим по индукции, возвращаем, красим в свободный цвет.

Лемма о триангуляции: доказательство

Доказательство.



- Отрезок между соседними вершинами не лежит в многоугольнике:

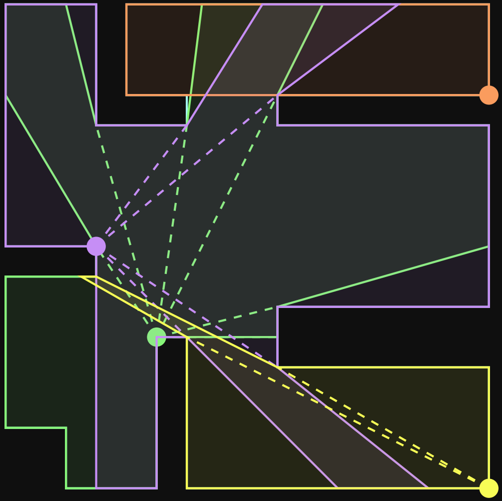


Задача о картинной галерее

Задача

Дана картинная галерея, план которой — многоугольник без самопересечений. Какое минимальное число охранников нужно поставить в точках галереи, чтоб они просматривали каждую точку?

Задача о картинной галерее: иллюстрация



Задача о картинной галерее

Теорема (Хватал)

Для произвольного n-угольника досаточно $\lfloor \frac{n}{3} \rfloor$ охранников, поставленных во внутренниих точках, чтоб охранникам были видны все внутренние точки n-угольника.

Доказательство.

По лемме строим разбиение галереи на треугольники так, что полученный граф раскрашивается в 3 цвета.

Из этих цветов выбираем тот, который встречается *не чаще других*, им раскрашиваем $\lfloor \frac{n}{3} \rfloor$ вершин.

Ставим охранников в вершины, окрашенные этим цветом.

□