

Вычислительная геометрия

Борис Золотов Матвей Магин

14 июня 2022 г.

Летняя школа МКН СПбГУ

Algorithms to get a feeling

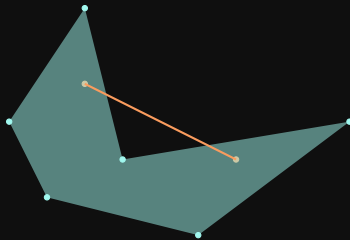
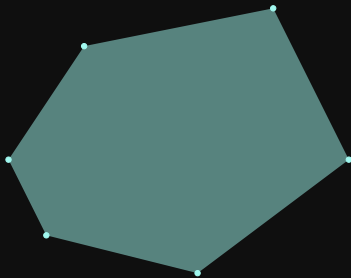
Выпуклая оболочка

Algorithms to get a feeling

Выпуклое множество

Определение

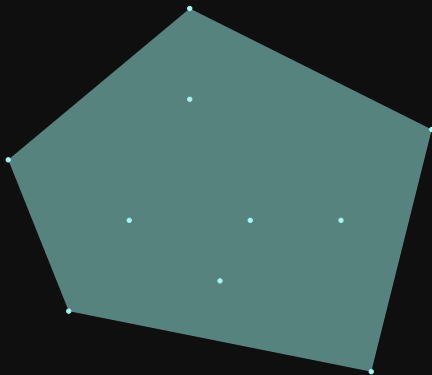
Множество S — **выпуклое**, если оно вместе с любыми двумя точками содержит отрезок между ними.



Выпуклая оболочка

Определение

Выпуклая оболочка $\text{CH}(S)$ 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

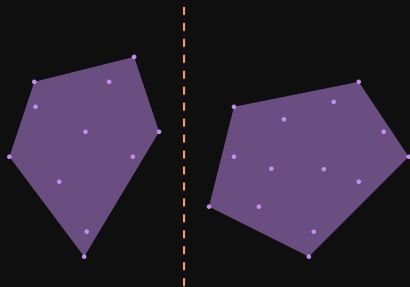
- Алгоритм Джарвиса – Selection Sort.
- Quick Hull – Quick Sort.
- Алгоритм «Разделяй и властвуй» – Merge Sort.

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

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

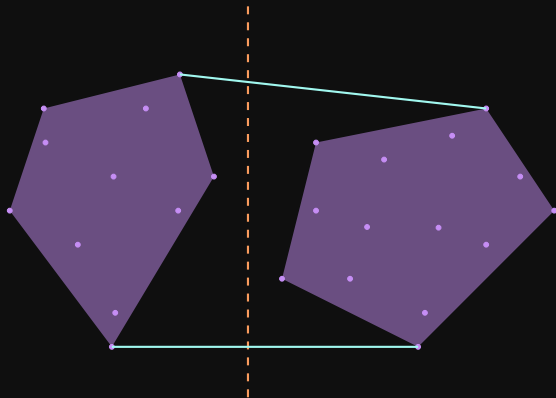
Алгоритм $D\&C$ для \mathcal{CH} : описание

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



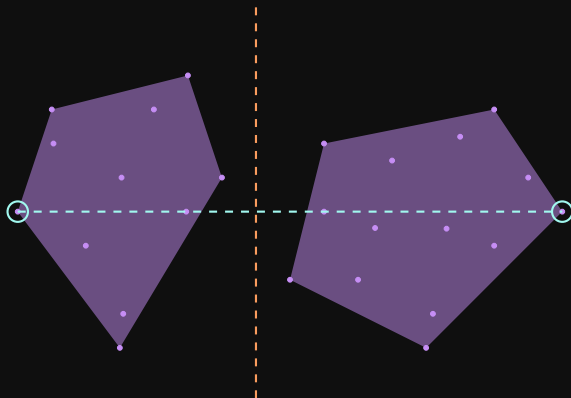
Алгоритм $D\&C$ для CH : слияние подзадач

Для слияния подзадач будем считать **верхнюю** и **нижнюю** касательные.



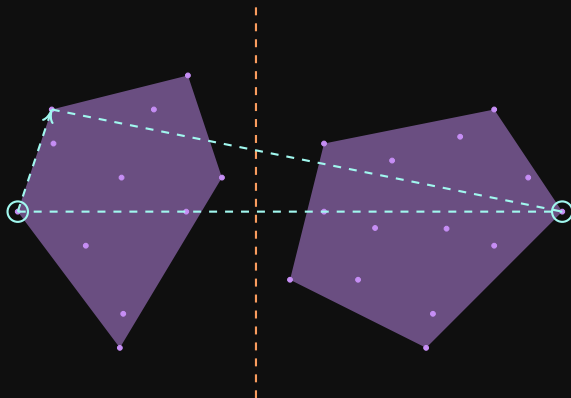
Алгоритм $D\&C$ для CH : верхняя и нижняя касательные

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



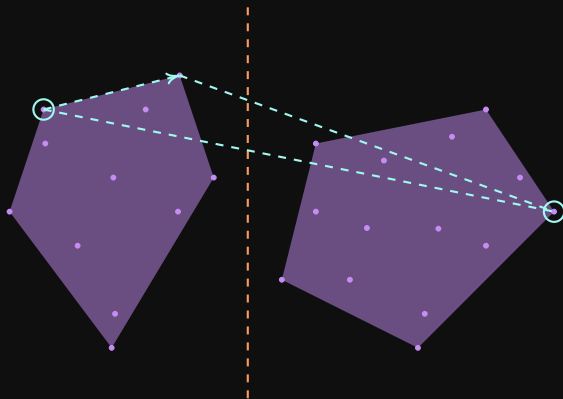
Алгоритм $D\&C$ для CH : верхняя и нижняя касательные

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



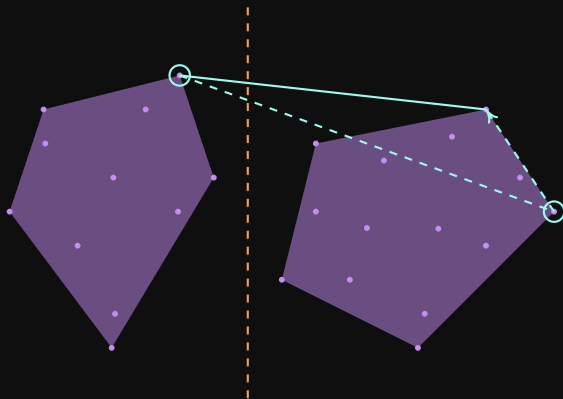
Алгоритм $D\&C$ для \mathcal{CH} : верхняя и нижняя касательные

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



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

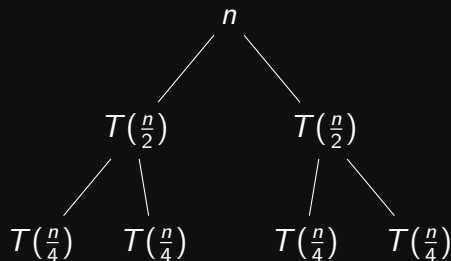
Идея вычисления: поднимаем тот конец отрезка, который можем поднять.



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

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

Will be upd...
