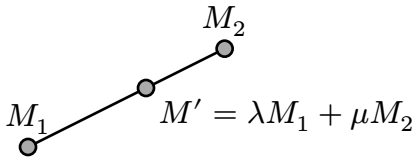
2.3 Выпуклые оболочки

Рассмотрим точечное множество $M = \{M_1, ..., M_n\}$

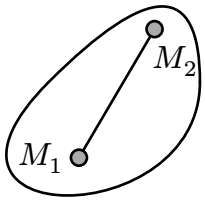
Mem. Выпуклое множество – такое множество, в котором для любых $M_i, M_j \in M$ отрезок M_iM_j тоже принадлежит множеству M

Nota. Также выпуклое множество – множество, где линейная (или аффинная) комбинация любых точек принадлежит множеству: $\forall M_{i_1}, ..., M_{i_k} \in M$

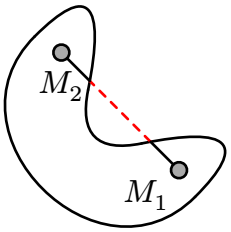
$\sum_{j=1}^{k} \lambda_j M_{i_j} \in M$ для всех $\lambda_i \geq 0$ таких, что $\sum_{i=1}^{k} \lambda_i = 1$



Ex.

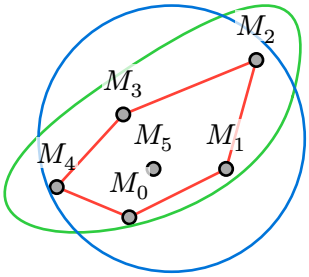


Выпуклое множество



Невыпуклое множество

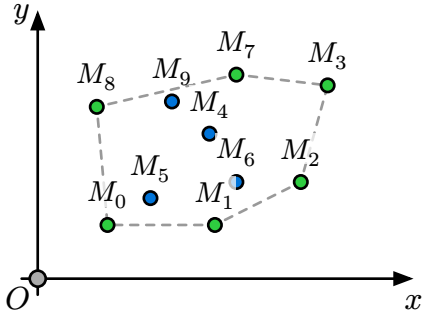
Для данного точечного набора существуют различные выпуклые множества, содержащие набор точек



Def. Наименьшее из выпуклых множеств V, содержащих данное множество M, называется выпуклой оболочкой $\operatorname{conv} M$ множества M

Nota.
1. Выпуклая оболочка пустого множества – пустое множество: $\operatorname{conv} \varnothing = \varnothing$
2. Выпуклая оболочка множества из точки – это множество из точки: $\operatorname{conv}\{M\} = \{M\}$
3. Выпуклая оболочка множества из двух точек – это отрезок: $\operatorname{conv}\{M_1, M_2\} = M_1M_2$
4. Выпуклая оболочка в общем случае $\operatorname{conv}\{M_i\}_{i=1}^{n}$ – это выпуклый многоугольник
5. Выпуклая оболочка выпуклого многоугольника – сам многоугольник

Выделим крайние и внутренние точки множества $M = \{M_i\}_{i=1}^{n}$: крайние точки – это те точки, что являются вершинами многоугольника, остальные считаем внутренние



Если найти все крайние, тогда построение $\operatorname{conv} M$ сводится к их упорядочиванию этих точек и построению ребер

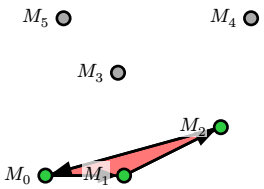
Критерий. Точка M не является крайней (то есть является внутренней) тогда и только тогда, когда существует треугольник $\triangle M_i M_j M_k$, для которого

$M \in \triangle M_i M_j M_k$ и $M \neq M_i, M_j, M_k$

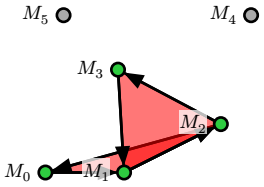
Действуя по критерию можно найти все внутренние точки, но алгоритм является очень затратным

Рассмотрим такую ситуацию:

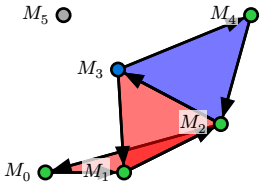
1. Проверяем ориентацию $\triangle M_0 M_1 M_2$, например, с помощью ориентированной площади $S_\triangle = \frac{1}{2}\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$



2. Если треугольник $\triangle M_0 M_1 M_2$ ориентирован против часовой стрелки, то переходим к $\triangle M_1 M_2 M_3$
3. Если треугольник $\triangle M_1 M_2 M_3$ ориентирован против часовой стрелки, то переходим к $\triangle M_2 M_3 M_4$



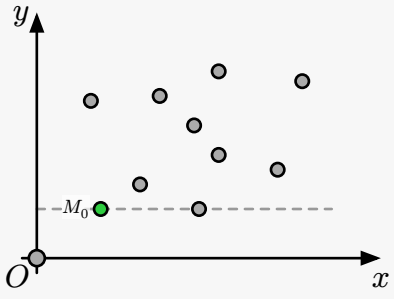
4. Если треугольник $\triangle M_2 M_3 M_4$ ориентирован в другую сторону, то считаем M_3 внутренней и рассматриваем треугольник $\triangle M_2 M_4 M_5$



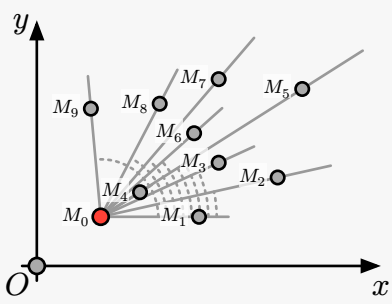
Итак, алгоритм Грэхема:

Дано $M = \{M_i\}_{i=1}^n$ в экранной системе координат

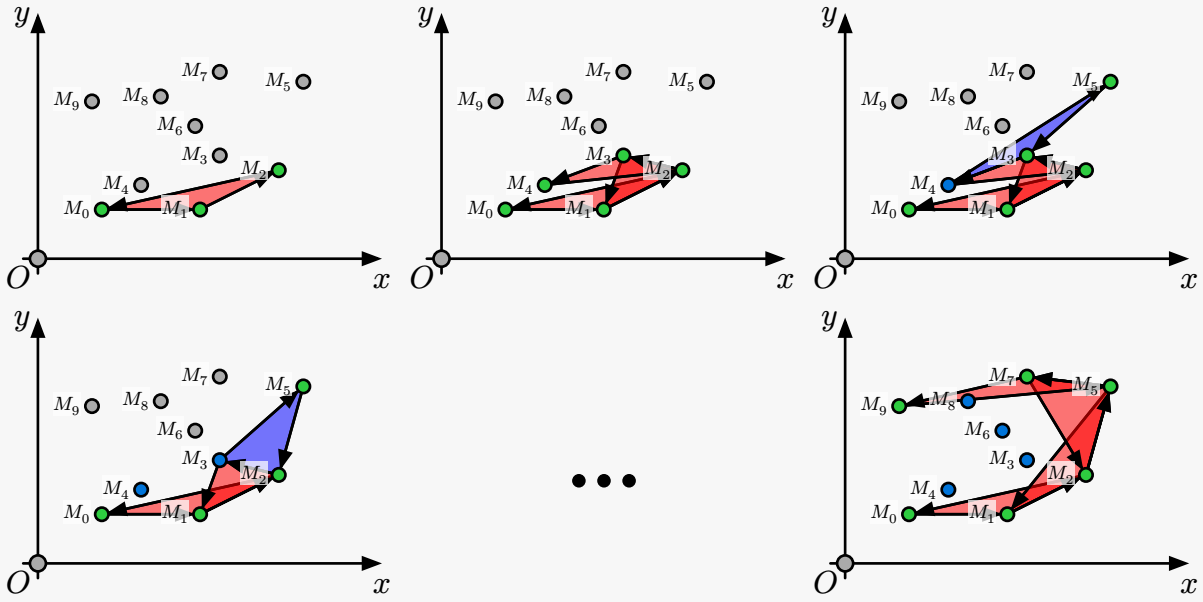
1. Отбираем точки $M_{\min y}$ из M так что $y_0 = \min\limits_{i=1..n} y_i$, далее находим точку M_0 из $M_{\min y}$ такую, что $x_0 = \min\limits_{i=1..n} x_i$ (она будет самой левой из самых нижних и являться крайней)



2. Далее упорядочиваем точки M_i по величине угла в M_0 между каждой точкой M_i и осью Ox



3. Определяем ориентацию троек $M_iM_{i+1}M_{i+2}$. Если точки идут по часовой стрелке, то исключаем M_{i+1}, повторяем с $M_{i-1}M_iM_{i+2}$ до тех пор, пока точки будут идти против часовой стрелки

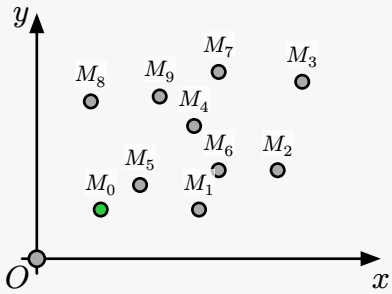


Такой алгоритм из-за сортировки углов работает за $O(n \log n)$, где $n = |M|$ – число точек во множестве

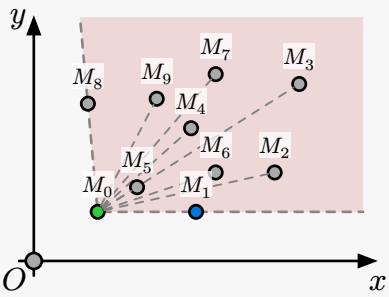
Другой метод, алгоритм Джарвиса (или алгоритм заворачивания подарка), работает так:

Дано $M = \{M_i\}_{i=1}^{n}$ в экранной системе координат

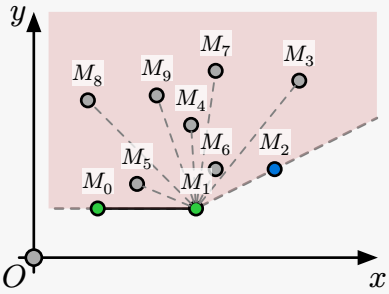
1. Находим самую левую точку M_0 из самых нижних (или другую такую, чтобы она была крайней)

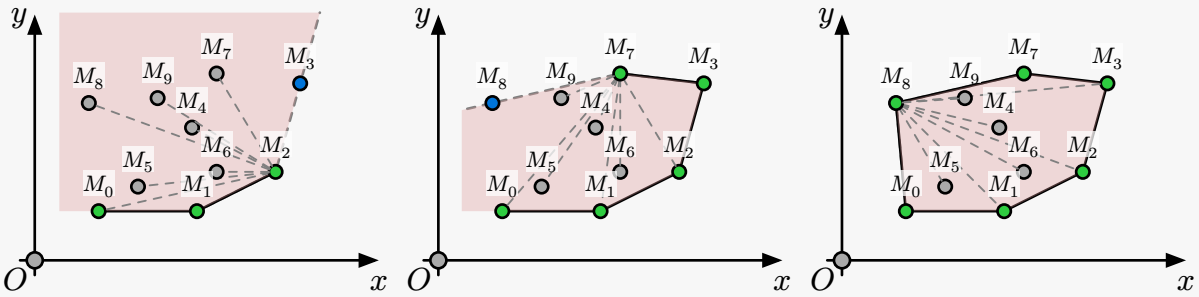


2. Относительно нее находим ту точку M_1, которая будет составлять минимальный угол между лучами M_0M_1 и осью Ox. Такая точка M_1 будет являться крайней



3. Повторяем до тех пор, пока не придем к M_0





Алгоритм Джарвиса имеет сложность $O(hn)$, где h – число вершин выпуклой оболочки (в данном примере – 6), а $n = |M|$ – число точек во множестве, поэтому он быстрее алгоритма Грэхема в тех случаях, когда $h < \log n$