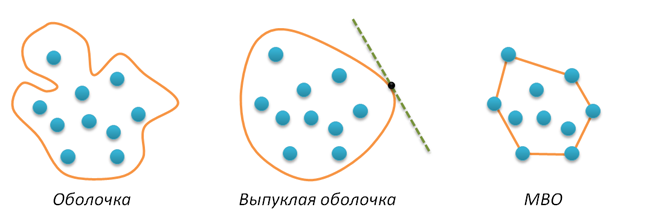
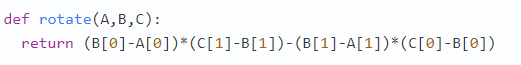
4.Понятие минимальной выпуклой оболочки **(МВО-минимальный выпуклая оболочка)**

Пусть на плоскости задано конечное множество точек A. Оболочкой этого множества называется любая замкнутая линия H без самопересечений такая, что все точки из A лежат внутри этой кривой. Если кривая H является выпуклой (например, любая касательная к этой кривой не пересекает ее больше ни в одной точке), то соответствующая оболочка также называется выпуклой. Наконец, минимальной выпуклой оболочкой (далее кратко МВО) называется выпуклая оболочка минимальной длины (минимального периметра). Я не проверял (похоже, это можно доказать от противного), но кажется очевидным, что минимальная оболочка просто обязана быть выпуклой. Все введенные понятия иллюстрируются следующим рисунком.

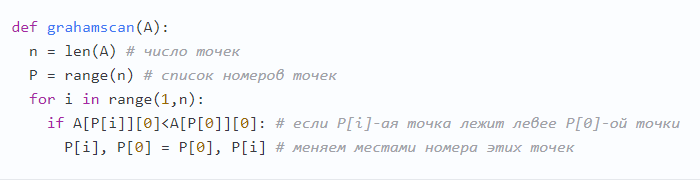


Главной особенностью МВО множества точек A является то, что эта оболочка представляет собой выпуклый многоугольник, вершинами которого являются некоторые точки из A. Поэтому задача поиска МВО в конечном итоге сводится к отбору и упорядочиванию нужных точек из A. Упорядочивание является необходимым по той причине, что выходом алгоритма должен быть многоугольник, т.е. последовательность вершин. Наложим дополнительно условие на порядок расположения вершин — направление обхода многоугольника должно быть положительным (напомню, что положительным называется обход фигуры против часовой стрелки).

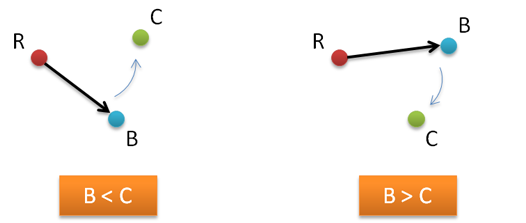
Задача построения МВО считается одной из самых простых задач вычислительной геометрии, для нее существует много различных алгоритмов. Ниже мы рассмотрим один такой алгоритм — Грэхема (Graham scan). Их описание иллюстрируется кодом на Питоне. Обоим алгоритмам потребуется функция rotate, побробно описанная в предыдущем моем [посте](http://habrahabr.ru/post/144571/). Напомню, что эта функция определяет, с какой стороны от вектора AB находится точка C (положительное возвращаемое значение соответствует левой стороне, отрицательное — правой). 

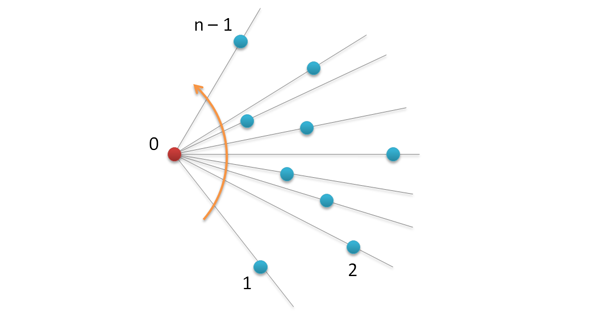
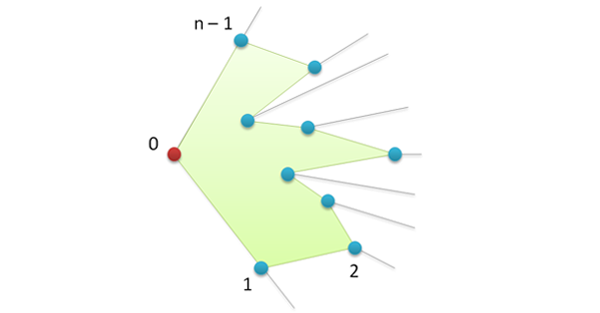
**Алгоритм Грэхема (Graham scan)**

Этот алгоритм является трехшаговым. На первом шаге ищется любая точка в A, гарантированно входящая в МВО. Нетрудно сообразить, что такой точкой будет, например, точка с наименьшей x-координатой (самая левая точка в A). Эту точку (будем называть ее стартовой) перемещаем в начало списка, вся дальнейшая работа будет производиться с оставшимися точками. По некоторым соображениям, исходный массив точек A нами меняться не будет, для всех манипуляций с точками будем использовать косвенную адресацию: заведем список P, в котором будут хранится *номера* точек (их позиции в массиве A). Итак, первый шаг алгоритма заключается в том, чтобы первой точкой в P оказалась точка с наименьшей x-координатой. Код:

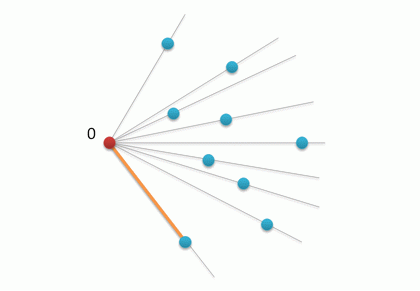


Второй шаг в алгоритме Грэхема — сортировка всех точек (кроме P[0]-ой), по степени их *левизны* относительно стартовой точки R=AP[0]. Будем говорить, что B<C, если точка С находится по левую сторону от вектора RB.

Для выпонения такого упорядочивания можно применять любой алгоритм сортировки, основанный на попарном сравнении элементов, например, быструю сортировку.

Результат сортировки можно проиллюстрировать следующим рисунком.  
  
Если мы теперь соединим точки в полученном порядке, то получим многоугольник, который, однако, не является выпуклым.  
  
Переходим к третьему действию. Все, что нам осталось сделать, так это срезать углы. Для этого нужно пройтись по всем вершинам и удалить те из них, в которых выполняется правый поворот (угол в такой вершине оказывается больше развернутого). Заводим стек S (реально список) и помещаем в него первые две вершины (они, опять же, гарантированно входят в МВО). Затем просматриваем все остальные вершины, и отслеживаем направление поворота в них с точки зрения последних двух вершин в стеке S: если это направление отрицательно, то можно срезать угол удалением из стека последней вершины. Как только поворот оказывается положительным, срезание углов завершается, текущая вершина заносится в стек.

В итоге в стеке S (который теперь можно рассматривать, как список) оказывается искомая последовательность вершин, причем в нужной нам ориентации, определяющая МВО заданного множества точек A.

  
Сложность первого и последнего шагов алгоритма является линейной по n (хотя в последнем случае имеется вложенный цикл, однако, каждая вершина внутри этого цикла ровно один раз заносится в стек, и не более одного раза оттуда удаляется), следовательно, сложность всего алгоритма определяется вторым шагом — сортировкой, именно поэтому сортировка вставкой оказывается не лучшим вариантом при больших n. Если ее заменить на быструю сортировку, то получим суммарную сложность алгоритма O(nlogn). Можно ли улучшить это время? Если алгоритм основан на попарном сравнении точек (как у нас), то доказано, что данная оценка в общем случае не улучшаема. С этой точки зрения алгоритм Грэхема оптимален. Тем не менее у него имеется не очень хорошая особенность — он не является адаптивным в том смысле, что не важно, сколько вершин в итоге войдет в МВО (три, пять, десять или n), все равно время будет линейно-логарифмическим. Такой адаптивностью обладает алгоритм Джарвиса, к рассмотрению которого мы плавно и переходим.