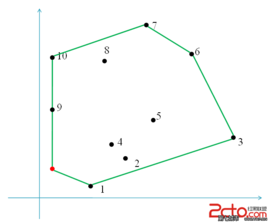
# 凸包

<https://blog.csdn.net/Bone_ACE/article/details/46239187>

凸包是计算几何学的概念，给定平面内的点集，构造一个凸多边形，使得所有点都在凸多边形内或者上，



## 枚举法：

两点确定一条直线，如果剩余的其它点都在这条直线的同一侧，则这两个点是凸包上的点，否则就不是。 时间复杂度n^3  
步骤：

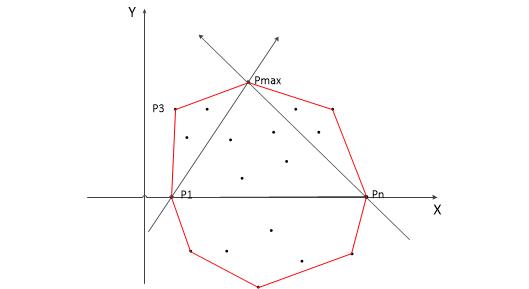
1. 将点集里面的所有点两两配对，组成 n(n-1)/2 条直线。
2. 对于每条直线，再检查剩余的 (n-2) 个点是否在直线的同一侧。判断同一侧转到是计算集合基本模板

## ****分治法:****

思路：应用分治法思想，把一个大问题分成几个结构相同的子问题，把子问题再分成几个更小的子问题……。然后我们就能用递归的方法，分别求这些子问题的解。最后把每个子问题的解“组装”成原来大问题的解。 时间复杂度：O(n㏒n)

这个算法要求点到线距离用来比较大小常数较大，所以不如后面的好用  
步骤：

1. 把所有的点都放在二维坐标系里面。那么横坐标最小和最大的两个点 P1 和 Pn 一定是凸包上的点（为什么呢？用反证法很容易证明，这里不详讲）。直线 P1Pn 把点集分成了两部分，即 X 轴上面和下面两部分，分别叫做上包和下包。
2. 对上包：求距离直线 P1Pn 最远的点，即下图中的点 Pmax 。
3. 作直线 P1Pmax 、PnPmax，把直线 P1Pmax 左侧的点当成是上包，把直线 PnPmax 右侧的点也当成是上包。
4. 重复步骤 2、3。
5. 对下包也作类似操作。



## Andrew扫描法：

一个很高效的离线算法，首先对点按照先x后y,升序排序

把凸包分为上半部和下半部分着求，对于上半部，用一个数组模拟栈，把前2个点(最左边2个)加入栈，之后按顺序枚举其他点，对于每个点c都拿它和栈顶2个元素比较，栈顶元素所构成的直线是ab。若c在ab上面，就将栈顶一个元素出栈，再做刚才这个比较操作，直到c和ab共线或者c在ab下面，就把c入栈。

下半部的凸包同理。

排序复杂度n\*logn,扫描是n的时间复杂度

* 在判断三点关系时最好用向量叉积法，这是计算几何基础模板，顺序任意，只要正负关系屡清楚，但是某些题对于共线的点也有要求，应该处理清楚

## Graham扫描法:

## Melkman算法: