



高级算法



分治

My bilibili Account: <https://space.bilibili.com/238037021>

欢迎关注!

概要

本文实现并解释了李传艺老师在第6周《高级算法》课上所提到的“分治求凸包”。

其中，代码通过了 P2742 [USACO5.1] 圈奶牛Fencing the Cows / 【模板】二维凸包

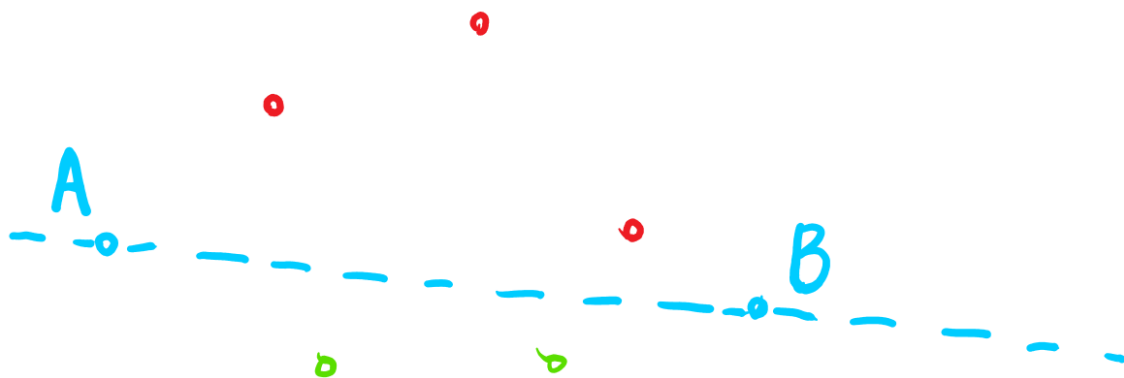
<https://www.luogu.com.cn/problem/P2742>

以证明其正确性。

提供了cpp源文件和伪代码。

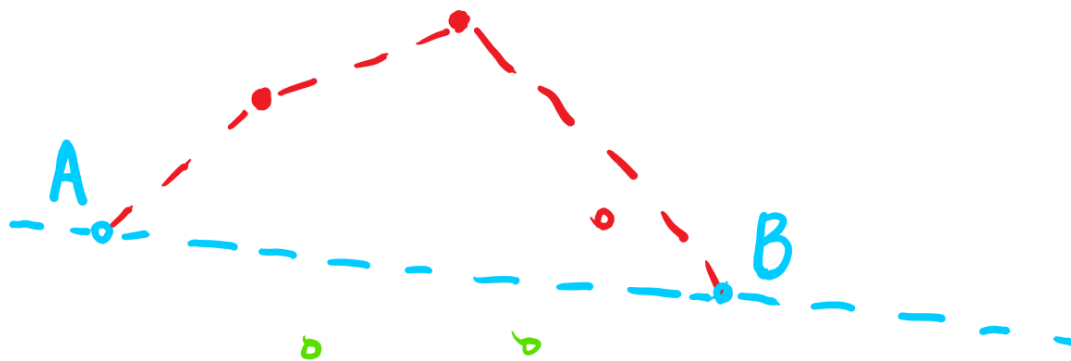
优雅实现

题目回顾:

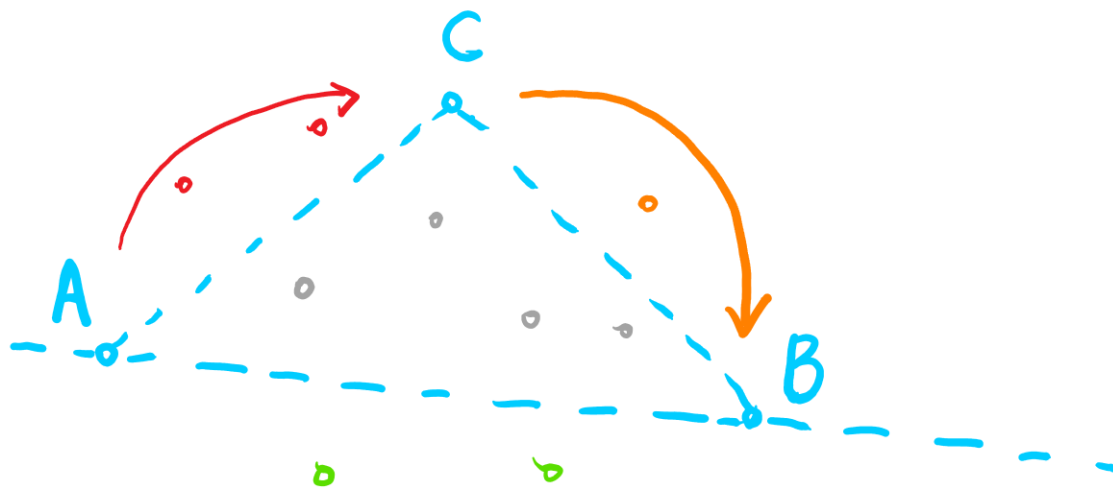


首先，我们选择了极左极右点，以他们的连线进行第一次分治，分为左侧点和右侧点两个子问题，它们可以用相同的方法解决。

现在我们要应用分治的问题是：给定一条直线 AB ，求解它左侧点的凸包，并以顺时针顺序列出这一段凸包的所有端点。我们不妨将 AB 左侧的所有凸包点称为 \widehat{AB} 。



我们将 \widehat{AB} 分治的方法是，找到一个距离 AB 最远的点 C ， C 必然作为凸包的一个端点，此时 \widehat{AB} 则被分解为了 $\widehat{AC} + C + \widehat{CB}$



Algorithm 1: *getConvex*

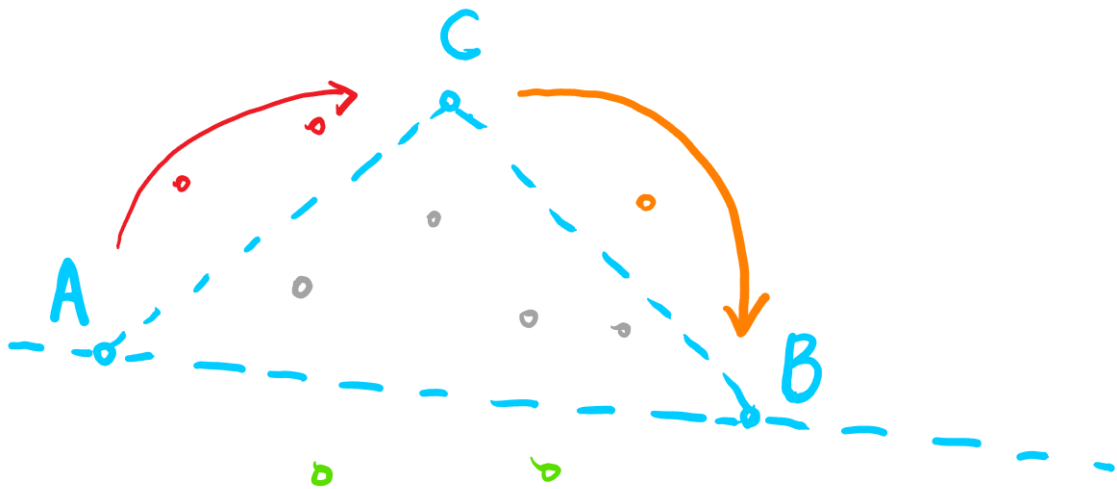
Input: 直线 AB ; 点集 S .

Output: 当函数返回时, 应已将所有凸包顶点压入答案队列 *queue*

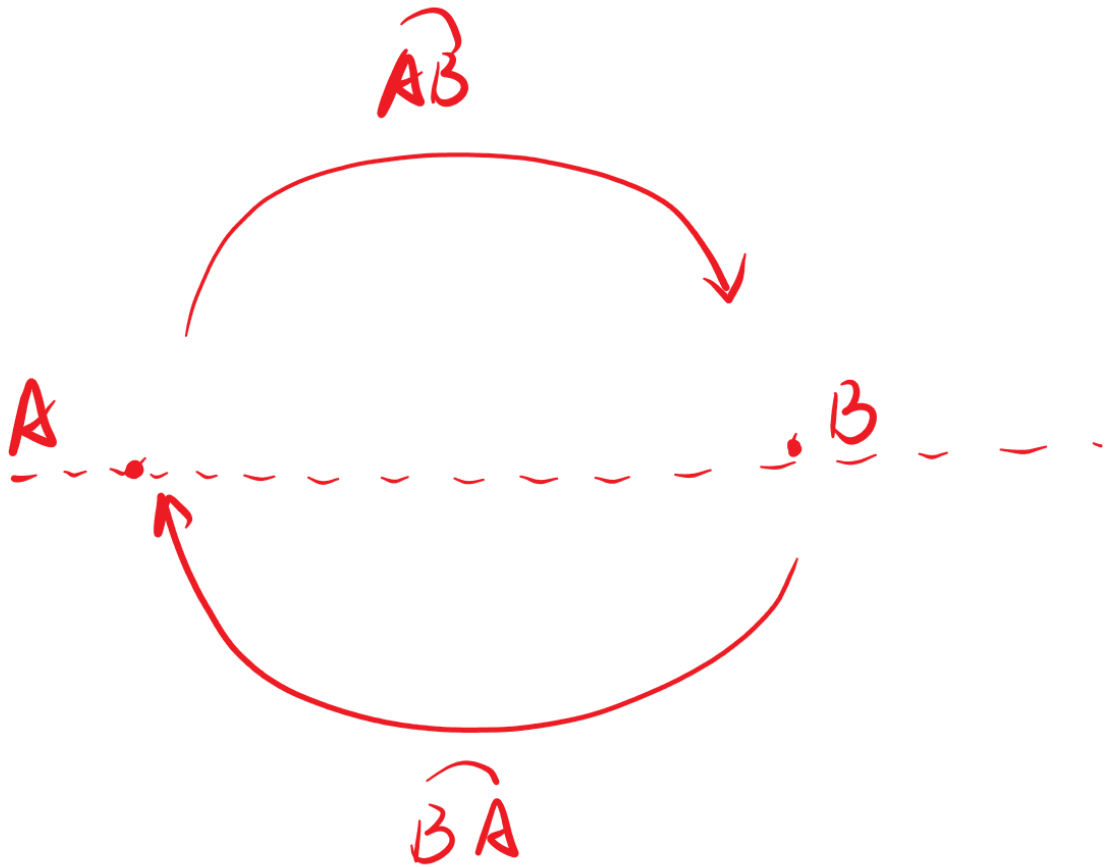
```
1 if  $S$  is empty then
2   | return
3  $C \leftarrow$  离  $AB$  最远点
4  $L \leftarrow \{\}$ 
5  $R \leftarrow \{\}$ 
6 foreach  $P_i \in S$  do
7   | if  $P_i$  在  $AC$  左侧 then
8     | add  $P_i$  to  $L$ 
9   | if  $P_i$  在  $CB$  左侧 then
10    | add  $P_i$  to  $R$ 
11  $getConvex(A, C, L)$ 
12 append  $C$  to  $queue$ 
13  $getConvex(C, B, R)$ 
```

我们声明了一个全局队列 $queue$, 函数直接向 $queue$ 中加点。为保证凸包顶点按照顺时针加入, 我们应总是按照如下顺序递归:

- 1.左递归 $getConvex(A, C, L)$
- 2.加入 C 点
- 3.右递归 $getConvex(C, B, R)$



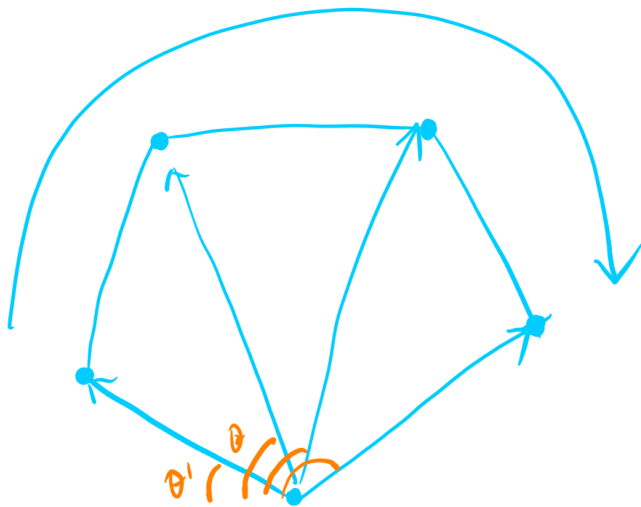
可以用数学归纳法证明其正确性: 假如 $getConvex(A, C, L)$ 能够正确顺时针加入 C 左侧所有点, $getConvex(C, B, R)$ 正确, 那么 $getConvex(A, B, S)$ 也能正确加入 \widehat{AB} 所有顶点。



对于最外侧的情况，我们也应当按如下顺序执行：

1. 加入A点。
2. 执行 $getConvex(A, B, \text{上半部分})$
3. 加入B点。
4. 执行 $getConvex(B, A, \text{下半部分})$

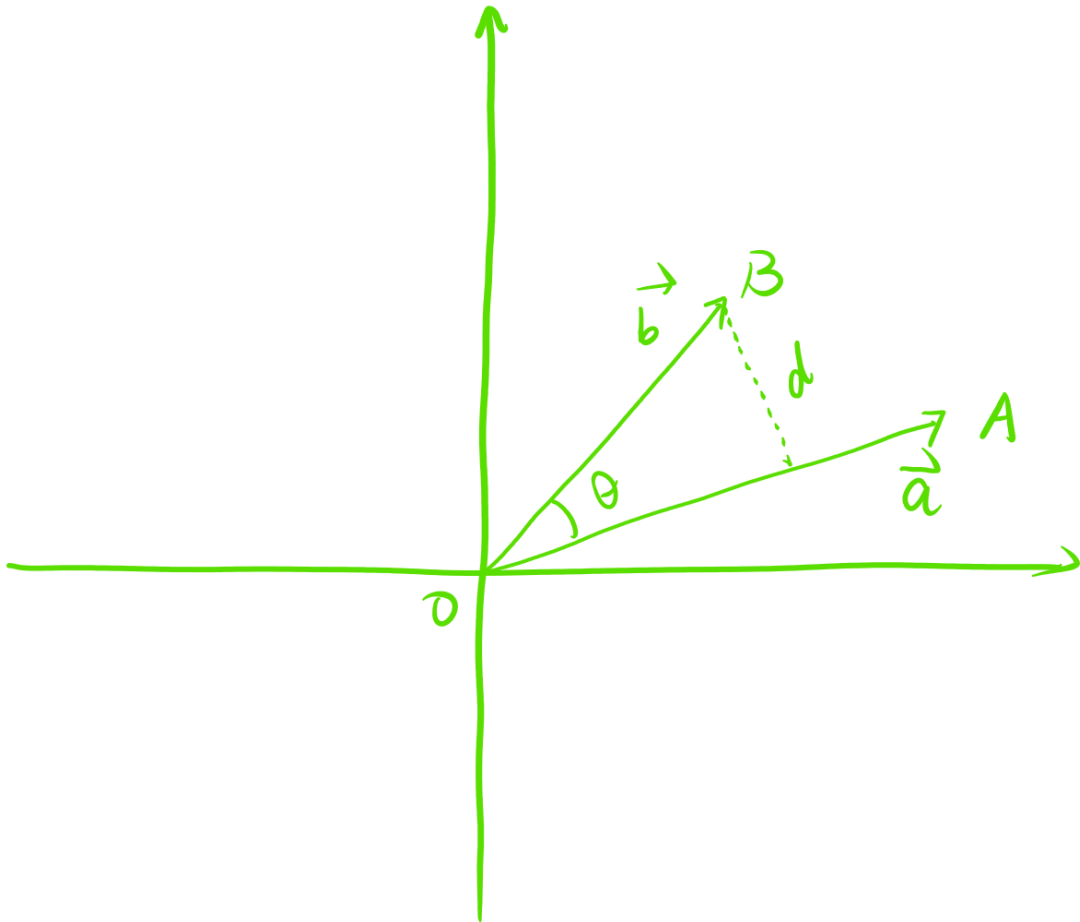
简单但实用的实现



$$\theta = \text{atan2}(y_1 - y_2, x_1 - x_2)$$

我们可以单纯地把所有顶点乱序放入`vector`中，找到其中`y`值最小的点，作为坐标系原点。由于凸包的特性，顺时针遍历所有点时，其极坐标角度单调减小。而极坐标可以用c++提供的函数`atan2(dy, dx)`计算。所以按极角排序可以得到逆时针排列的顶点。

如何用叉积快速判断点在直线的左右侧/到直线的距离



两向量的叉积 $Cross(\vec{a}, \vec{b}) = |\vec{a}||\vec{b}|\sin\theta = a.x * b.y - a.y * b.x = |\vec{a}|d$

其中 θ 是 \vec{a} 旋转到 \vec{b} 的角度，而 d 是点 B 到直线 OA 的距离

判断在直线左右的方法是，当 B 在 OA 左侧时，叉积为正，反之叉积为负。

在分治过程中，我们需要得到离 AB 最远的点。实际上，我们并不需要计算具体距离。此时 $|\vec{AB}|$ 是定值，两向量 \vec{AB}, \vec{AP} 的叉积 $Cross(\vec{AB}, \vec{AP}) = |\vec{AB}|d$ ，与 d 正相关。我们只要求叉积最大的点即可。