

برای اضافه کردن نقطه جدید، باید نقاطی را بیابیم که در پوسته ایجاد شده به این نقطه متصل اند. این کار را با دانستن این نکته که اگر دو نقطه جزئی از پوسته محدب باشند، آنگاه تمام نقاط مرزی این پوسته در طرف دیگر خط گذرنده از این دو نقطه خواهند بود انجام خواهیم داد.

همان طور که در صورت سوال نیز یادآوری شده است، پس از اضافه کردن این نقطه ممکن است تعدادی از نقاط درون این پوسته قرار گیرند که باید آن ها را از لیست نقاط مرزی حذف کنیم. این کار را با بررسی نقاط دیگر این پوسته انجام می دهیم به طوری که اگر p_1 و p_2 نقاطی باشند که در قسمت قبل پیدا کرده ایم، آنگاه اگر یکی از نقاط مرزی، سمت راست خط گذرنده از این دو نقطه باشد باید آن را حذف کنیم زیرا درون مثلی از نقاط مرزی خواهند بود.

Algorithm 1 GetOrientation

Require: Points P_1, P_2 and Q

Ensure: Orientation of Q with respect to P_1 and P_2

- 1: Right or top
- 2: Bottom or left
- 0: In the same line

$A = P_1P_2$

$B = P_1Q$

return $sign(A \times B)$

Algorithm 2 IsBoundaryLine

Require: Points Q_1, Q_2 and List Of Points $P = \{p_1, p_2, \dots, p_n\}$

Ensure: Boolean representing if Q_1Q_2 is a boundary line or not

orientation = GetOrientation(Q_1, Q_2, P_1)

for each point p in P **do**

if orientation * GetOrientation(Q_1, Q_2, p) < 0 **then**

return false

return true

Algorithm 3 GetConvexHull

Require: A list of points $P = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$

Ensure: Convex Hull of P

```
P = sortByX(P)
boundaryPoints = {p1, p2, p3}
for each p in P from p4 to pn do
    connectedPoints = {}
    for each point b in boundaryPoints do
        if IsBoundaryLine(p, b) then
            connectedPoints.append(b)
    for each point c in boundaryPoints do
        orientation = GetOrientation(connectedPoints[0], connectedPoints[1],
        c)
        if c is not in connectedPoints and orientation > 0 then
            boundaryPoints.remove(c)
    boundaryPoints.append(p)
return boundaryPoints
```
