تقسیم و غلبه - بخش سوم

Divide and Conquer III

پارادایم تقسیم و غلبه

Divide the problem into a number of subproblems that are smaller instances of the same problem.

Conquer the subproblems by solving them recursively. If the subproblem sizes are small enough, however, just solve the subproblems in a straightforward manner.

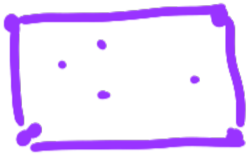
Combine the solutions to the subproblems into the solution for the original problem.

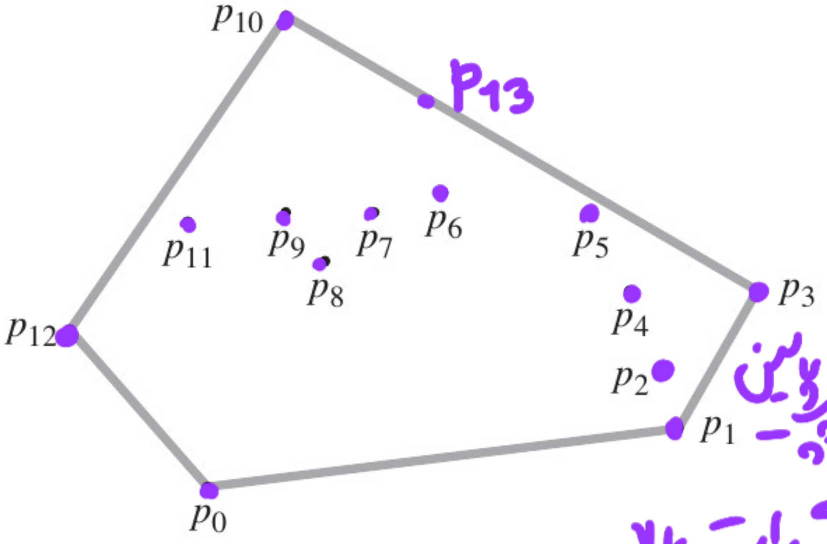
- Tournament Scheduling ← برنامه ریزی تورنمنت
- Convex Hull ← پوشش محدب

پوشش محدب

- Convex hull of a set ***Q*** of points, denoted by **CH**(***Q***), is
  - the smallest convex polygon ***P***
  - for which each point in ***Q*** is
  - either on the boundary of ***P*** or in its interior.

$p_i : (x_i , y_i)$





ولی بدنبال
جعبه محدودکننده
نیستیم

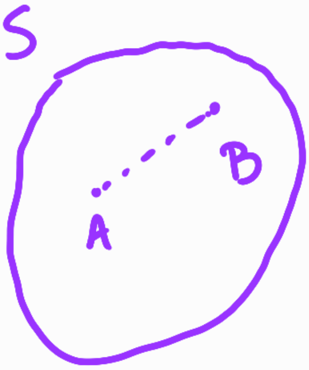
Boundin Box
جعبه محدودکننده
(مسئله ساده) ←

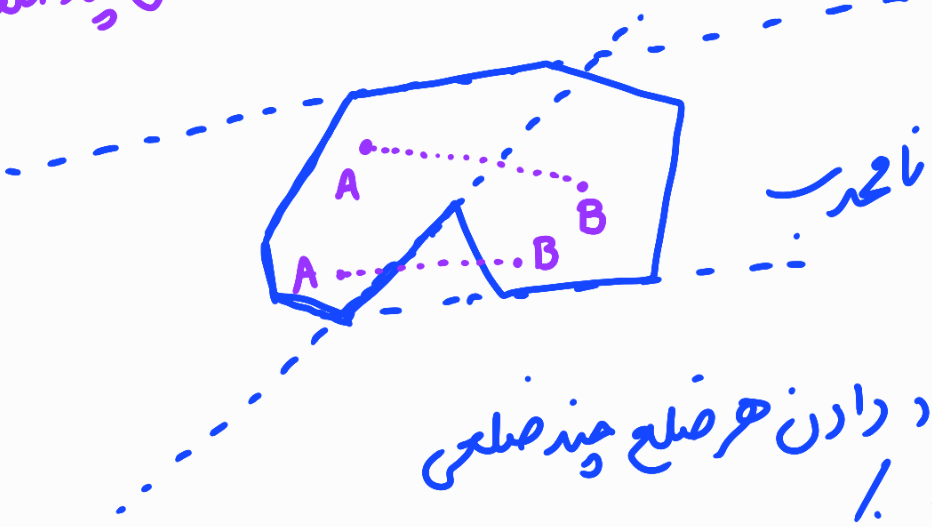
گوشه چپ پایین $(\min_i x_i , \min_i y_i)$

گوشه راست بالا $(\max_i x_i , \max_i y_i)$

چندضلعی محدب

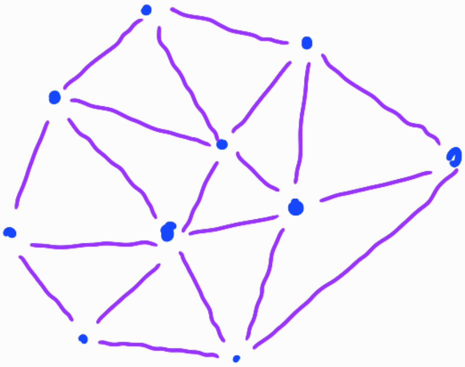
- پاره خط واصل بین هر دو نقطه داخل چندضلعی، از چندضلعی خارج نشود ⇐ محدب





- اگر امتداد دادن هر ضلع چندضلعی آن را قطع نکند ⇐ محدب

بسیاری از مسائل هندسه محاسباتی کاهش پذیر هستند به مسئله پوسش محدب



- مسئله مثلث بندی دلونی

- مسئله ناحیه بندی ورونی

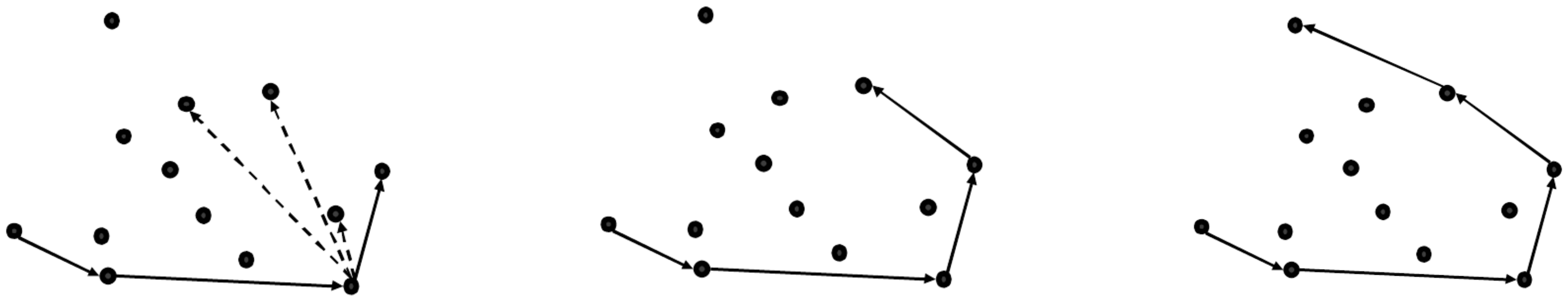
- ....

Q: مجموعه n نقطه

پوشش محدب

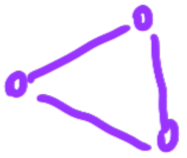
$C \subseteq Q$ C پوشش محدب

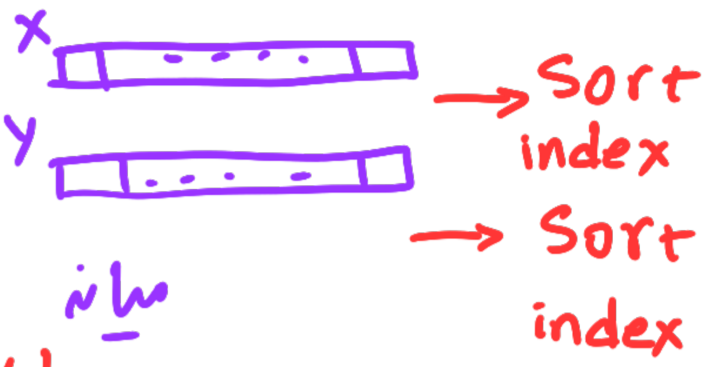
- There are many algorithms for computing the convex hull:
  - Brute Force: $O(n^3)$ ✓
  - Gift Wrapping: $O(n^2)$
  - Quickhull: $O(n\log n)$ – $O(n^2)$
  - **Divide and Conquer**: $O(n\log n)$ ✓



یک الگوریتم تقسیم و غلبه برای پیدا کردن پوشش محدب

$n=3$ ad-hoc





$L = x_{i_{\lceil n/2 \rceil}}$ میانه

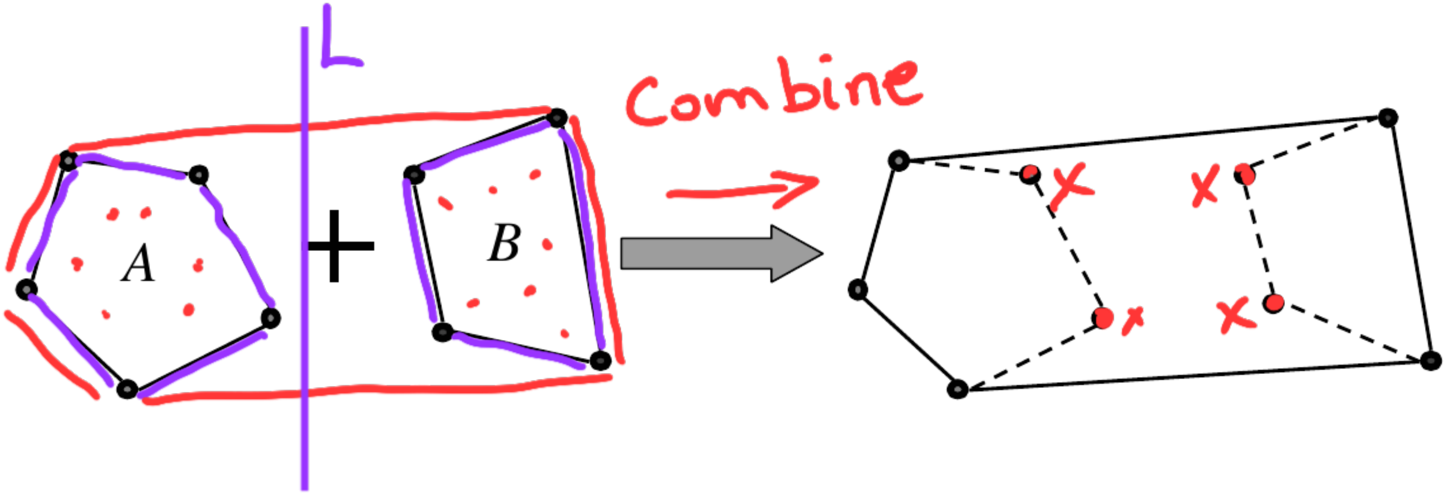
$x_{i_1} \leq x_{i_2} \leq \cdots \leq x_{i_n}$

Divide and conquer

✓ 1. Divide the n points into two halves.

✓ 2. Find convex hull of each subset.

✓ 3. Combine the two hulls into overall convex hull.



Combine

ترکیب پوشش‌های محدب

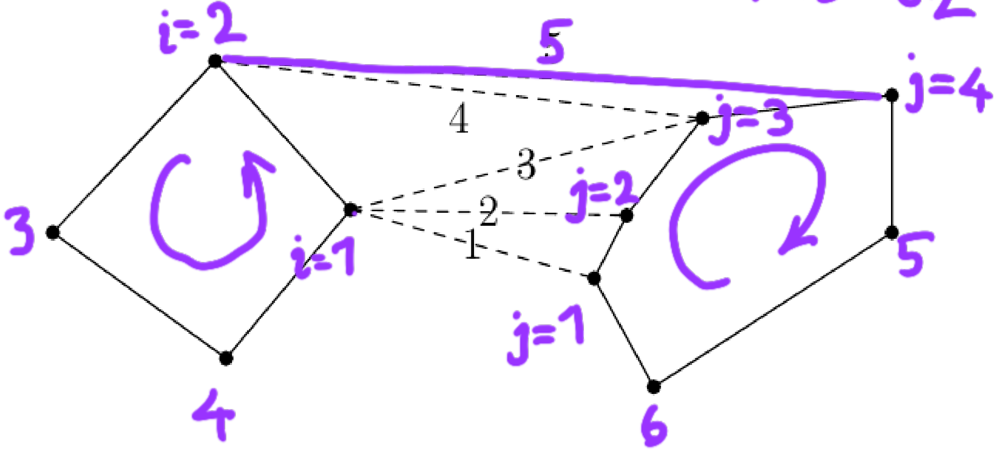
- Merging Hulls,
  - "*Stitch*" two hulls
  - مماس
  - Need to find the *tangent* joinng segments

لیست‌هایی که چندضلعی محدب را نشان می‌دهد

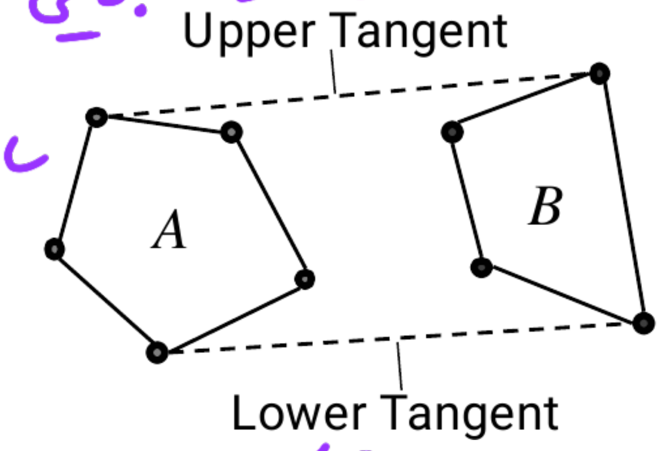
$O(n)$

List l_1 : پادساعتگرد

List l_2 : ساعتگرد



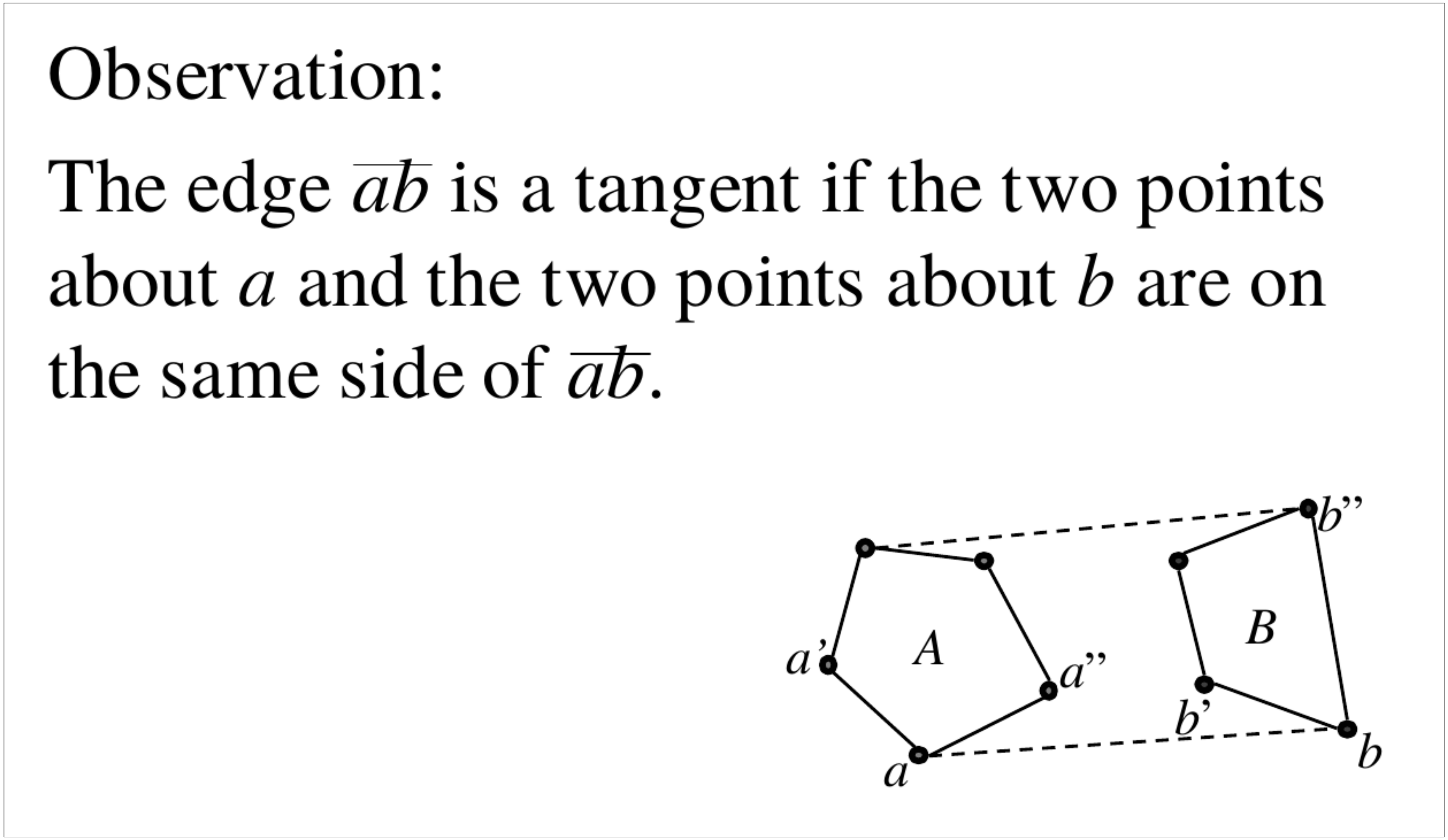
مماس مشترک بالایی



مماس مشترک پایینی

ترکیب پوشش‌های محدب

- Find a <u>*tangent*</u> joinng segment

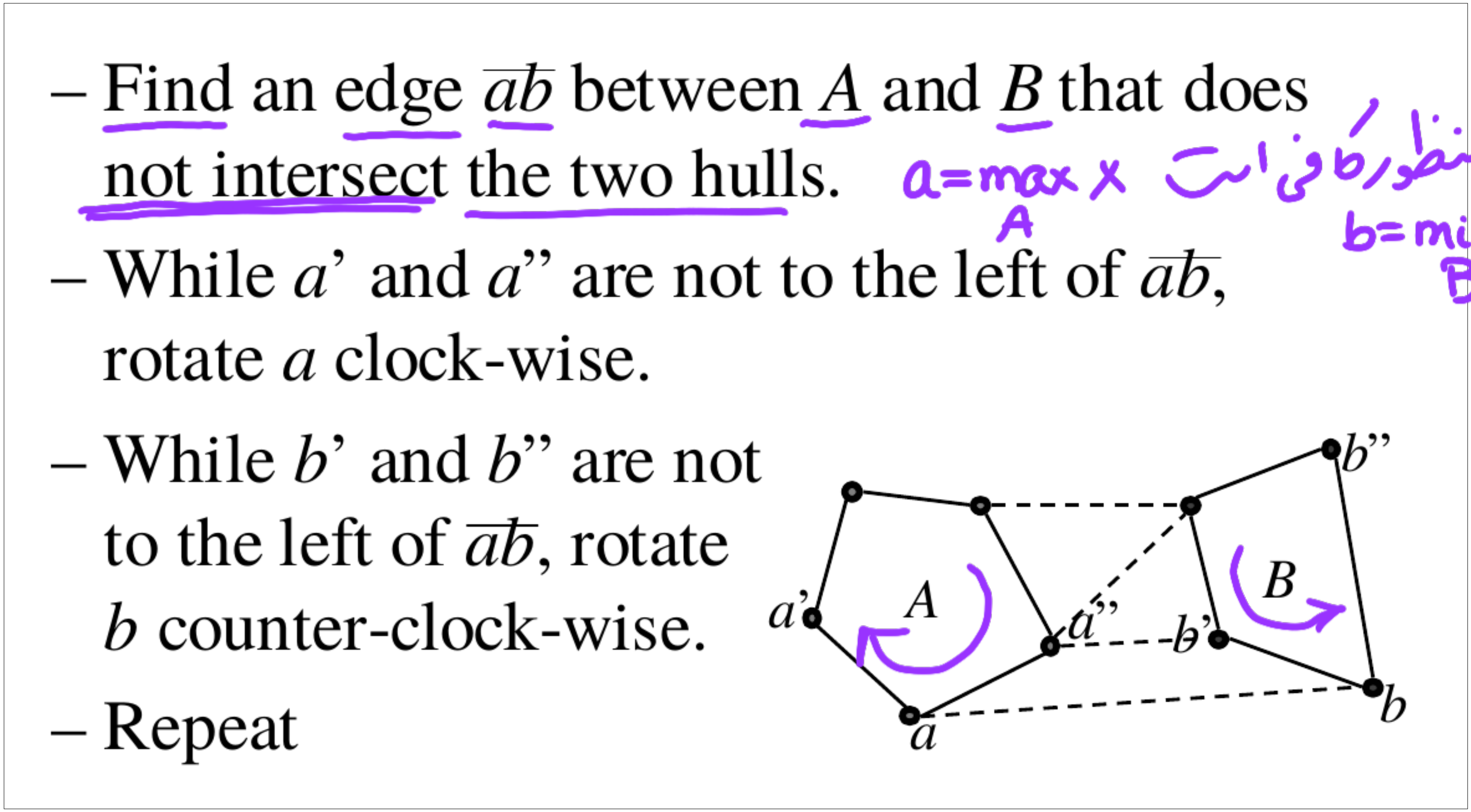


Observation:

The edge $\overline{ab}$ is a tangent if the two points about a and the two points about b are on the same side of $\overline{ab}$.

ترکیب پوشش‌های محدب

- “*Stitch*” algorithm



– Find an edge $\overline{ab}$ between *A* and *B* that does not intersect the two hulls.

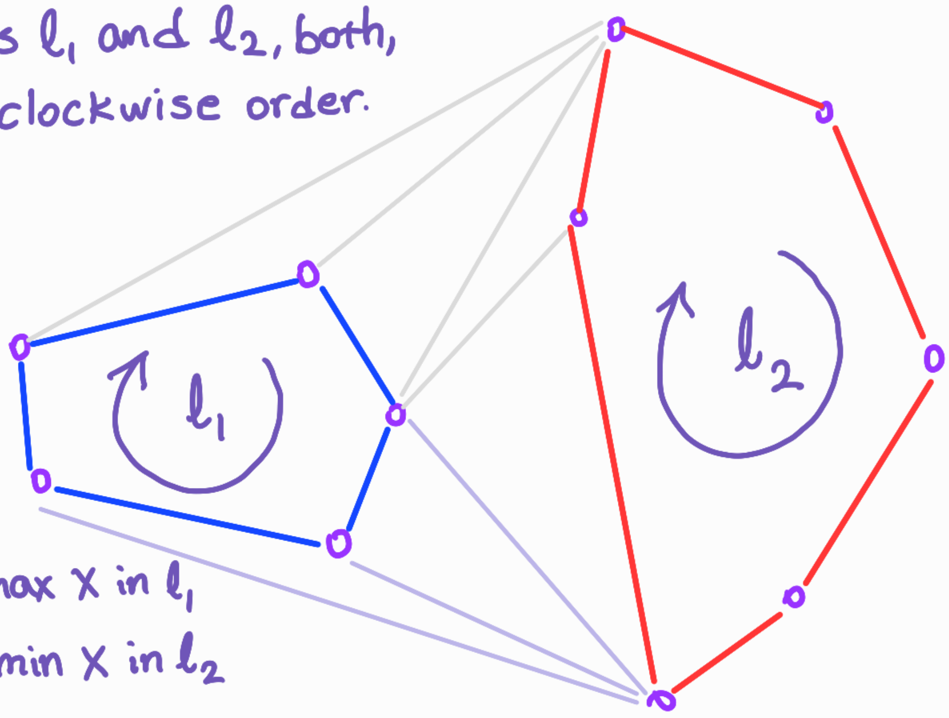
برای این منظور کافی است $a = \max_{A} x$ $b = \min_{B} X$

– While *a*’ and *a*” are not to the left of $\overline{ab}$, rotate *a* clock-wise.

– While *b*’ and *b*” are not to the left of $\overline{ab}$, rotate *b* counter-clock-wise.

– Repeat

Suppose lists l_1 and l_2, both, are sorted in clockwise order.



```
a = Point of max X in $l_1$
b = Point of min X in $l_2$
s1 = +1
s2 = +1
while s1==+1 OR s2==+1
    c = b.next
    s1 = sign(crossproduct($\vec{ab}$ , $\vec{ac}$))
    if s1 == +1
        b = c
    else
        c = a.prev
        s2 = sign(crossproduct($\vec{bc}$ , $\vec{ba}$))
        if s2 == +1
            a = c
return a,b // "upper joint tangent"
```

swap the role of "next" and "prev" in above code and replace every +1 with -1, then Do the same to find "Lower joint tangent"