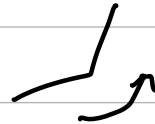


Convex Hull

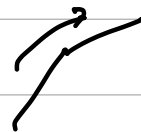
CCW

점 $A(x_1, y_1)$ $B(x_2, y_2)$ $C(x_3, y_3)$

A와 B의 기울기 < B와 C의 기울기 : 좌회전



A와 B의 기울기 > B와 C의 기울기 : 우회전



A와 B의 기울기 = B와 C의 기울기 : A, B, C는 한 직선에 존재

$$\begin{aligned} \text{좌회전} & 0 < \frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1} \\ \text{우회전} & 0 > \end{aligned}$$

$(y_2 - y_1)(y_3 - y_2) - (x_3 - x_2)(y_2 - y_1)$ 가 CCW 이고, $\begin{cases} \text{양수면 좌회전} \\ \text{음수면 우회전} \end{cases}$ 이다!

Brute-Force-ish $O(n^3)$

모든 선분마다 Convex Hull 의 일부분인지 확인한다.

선분에 대해 모든 점들이 우회전하거나 좌회전하면, Convex Hull에 있다고 판단.

Brute-Force-ish $O(n^2 \log n)$

점이 Convex Hull에 포함되는지 판단. 선분이 존재하는 각도가 180 이하여야 함.

각도 sorting 이 필요하므로 총 $O(n^2 \log n)$ 이 걸린다.

Package Wrapping $O(n^2)$

Convex Hull 위에 있는 것이 보장된 2차료가 가장 작은 점 위에서 시작.

나머지 모든 점들에서부터 가장 오른쪽 (또는 왼쪽)의 점을 찾고,

그 점에서부터 가장 오른쪽 (또는 왼쪽)의 점을 반복해서 찾는다.

Graham Scan $O(n \log n)$

Convex Hull 위에 있는 것이 보장된 2차료가 가장 작은 점 위에서 시작.

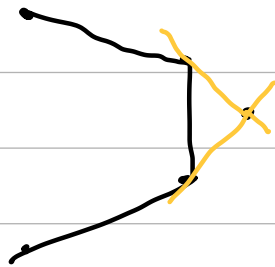
이 점을 기준으로 모든 점들에 대해 각도 순으로 정렬.

그 순서대로 Convex Hull에 넣고, CCW를 통해 좌회전하는지 확인한다.

우회전 한다면, 그 전에 추가했던 점들을 하나씩 제거하여 좌회전하게 만든다.

Plane Sweeping $O(n \log n)$

right hull 만 생각하자.



정답을 찾는다. 어떻게?

등고 나눌 때,



들어가는 점의 위쪽 점에 대해서는 우회전

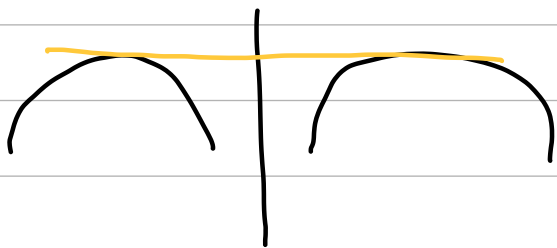
아래쪽 점에 대해서는 좌회전

Dynamic Case

새 점이 추가될 때, Convex Hull 안에 있는지 밖에 있는지 판단.

Divide and Conquer $O(\log^2 n)$

upper hull 판 생각해 보자.

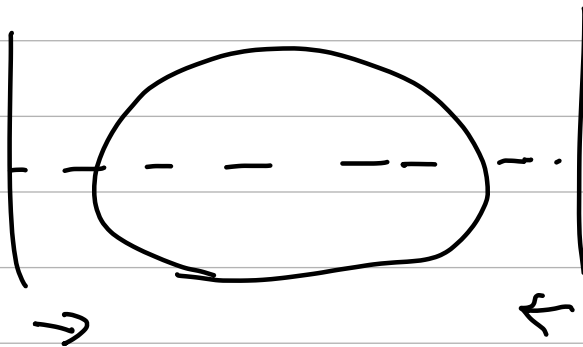


공동 점선을 찾아야 된다.



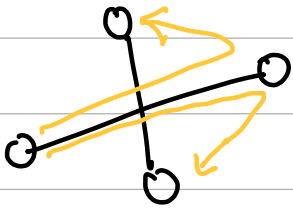
안 쪽을 때 가리 변좌아가며 확인.

Farthest Point

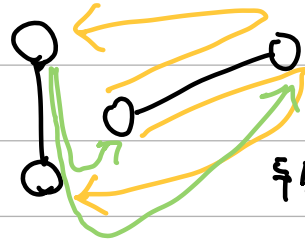


이런 n 개의 각도마다
다 해보면 됨. $\rightarrow O(n)$

선분의 교차 판별



그런데 이럴 때는?



두 선분에 대해 모두 확인함.



이럴 땐 위치 확인 필요!