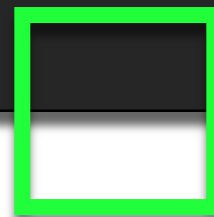


#고급반  
#6주차

# 볼록껍질

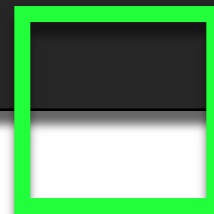
T. 장형준  
Asst. 장형준



#1

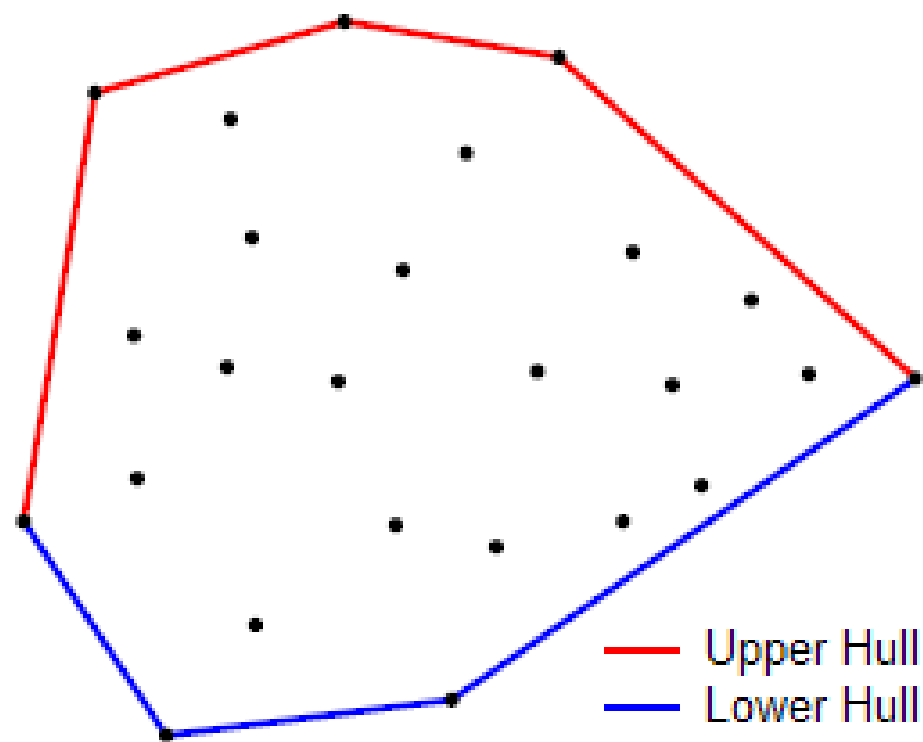
# 개념 설명

- 블록цеп질을 왜 알아야할까 -

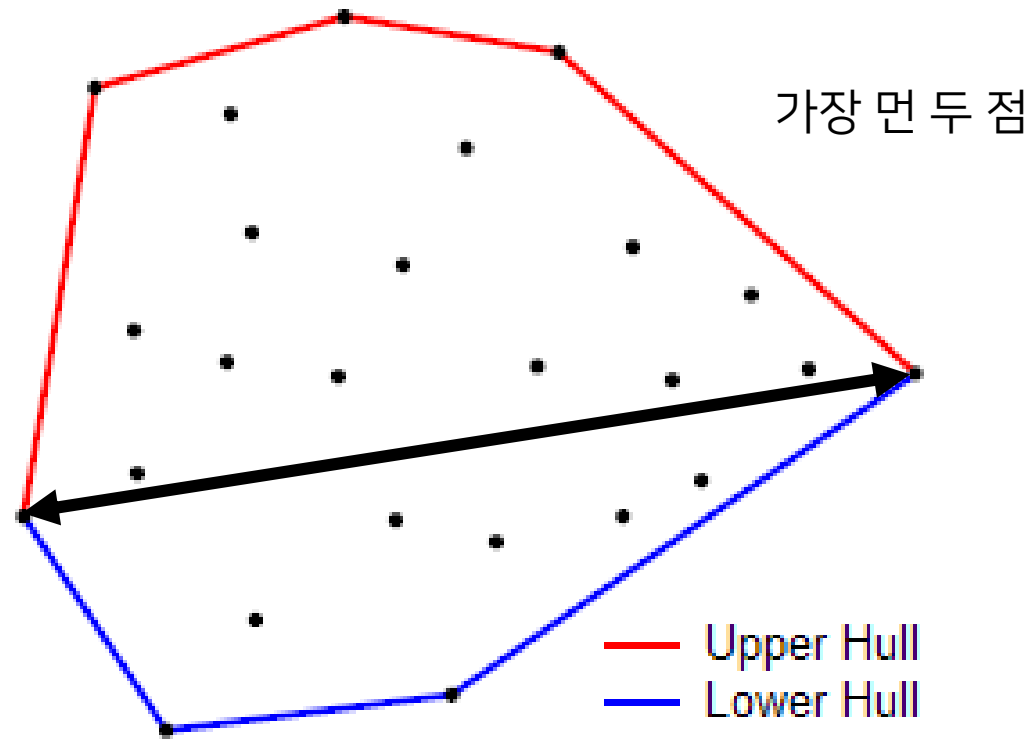


ALPHA  
The algorithm club.

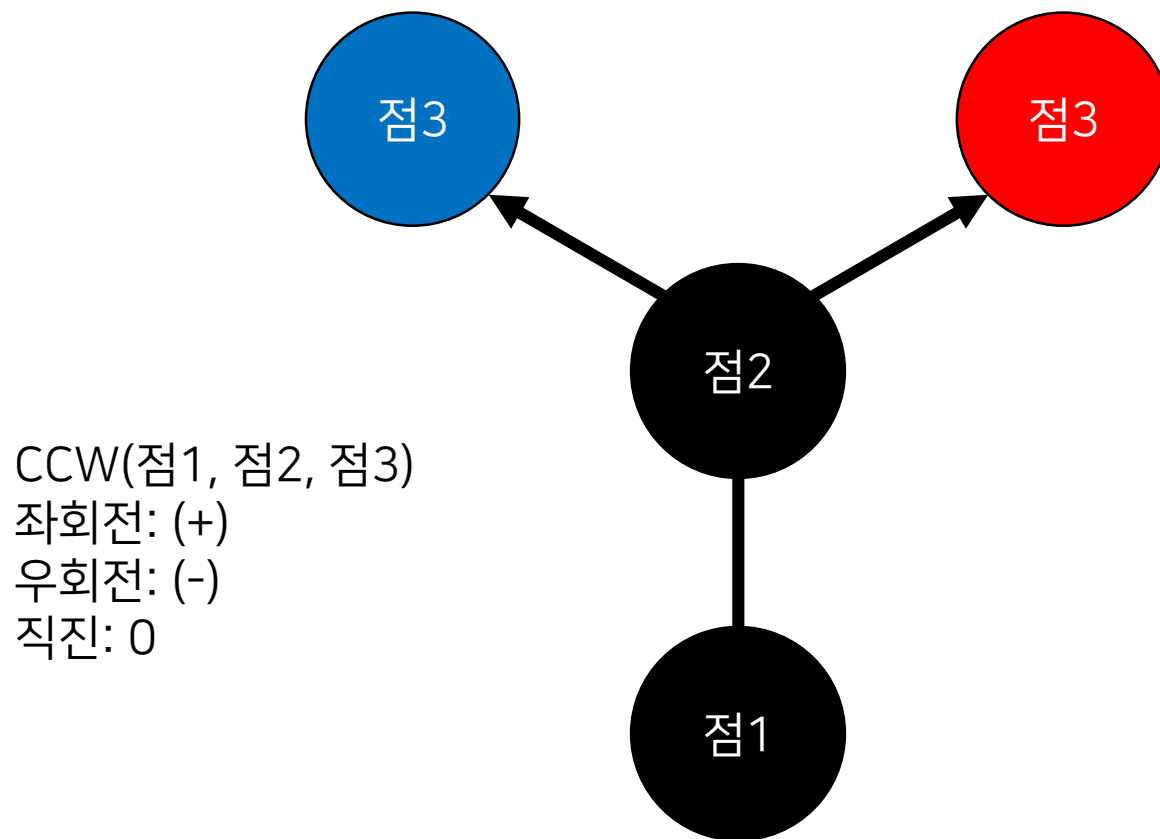
# 볼록 껍질이란?



# 어디에 쓰이나요?

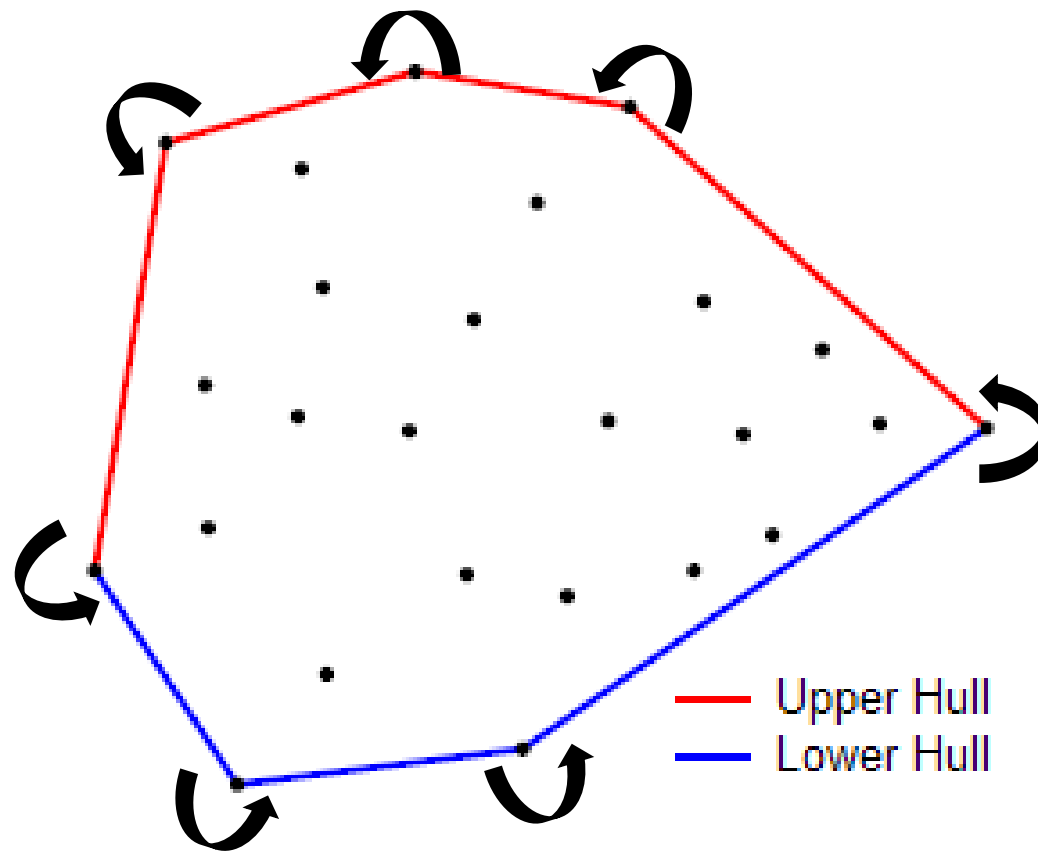


# 볼록한 걸 어떻게 알죠?



# 볼록한 걸 어떻게 알죠?

계속 좌회전(+)

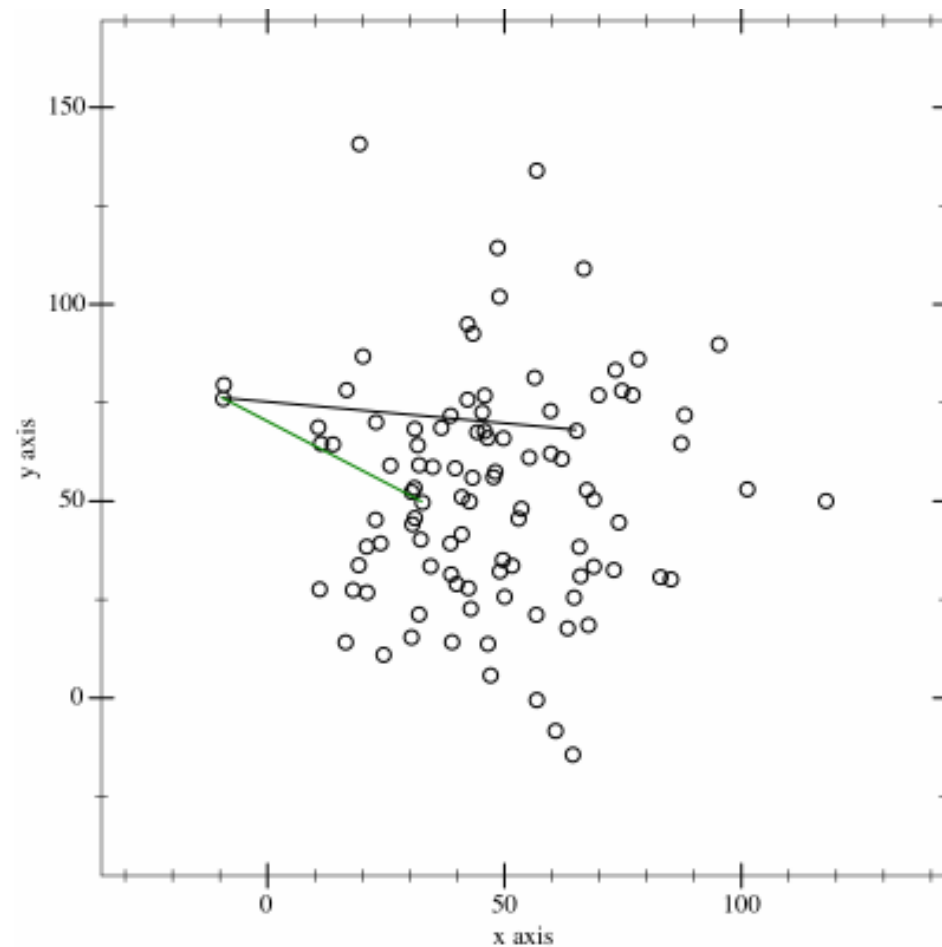
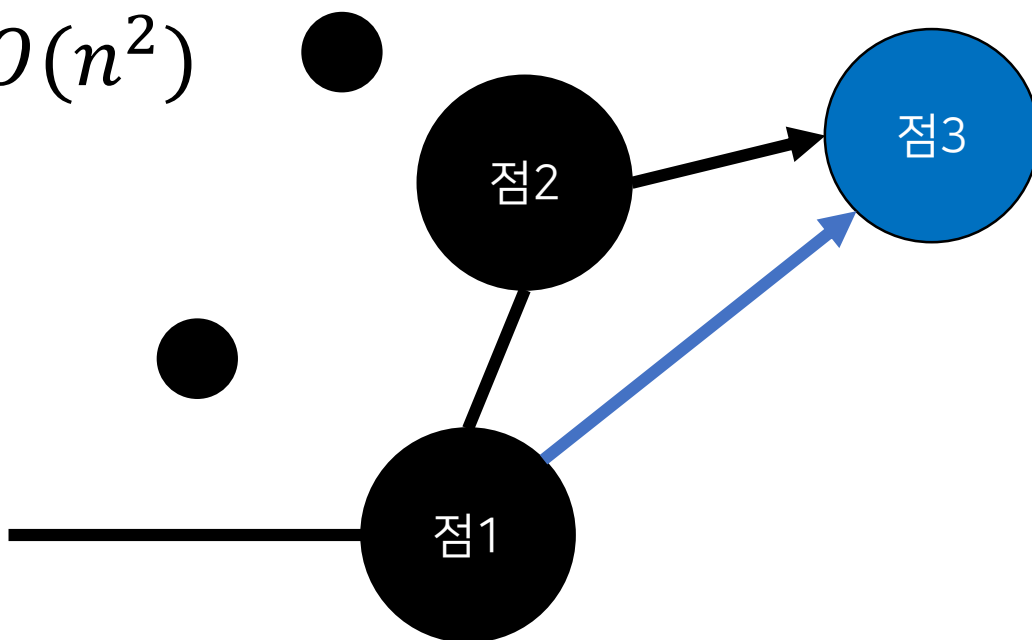


# Naive!

판서로 내용 보강하겠습니다!

가장 끝에 있는 점은 무조건 볼록껍질에 포함된다.  
규칙에 맞지 않으면 뺀다.

$O(n^2)$

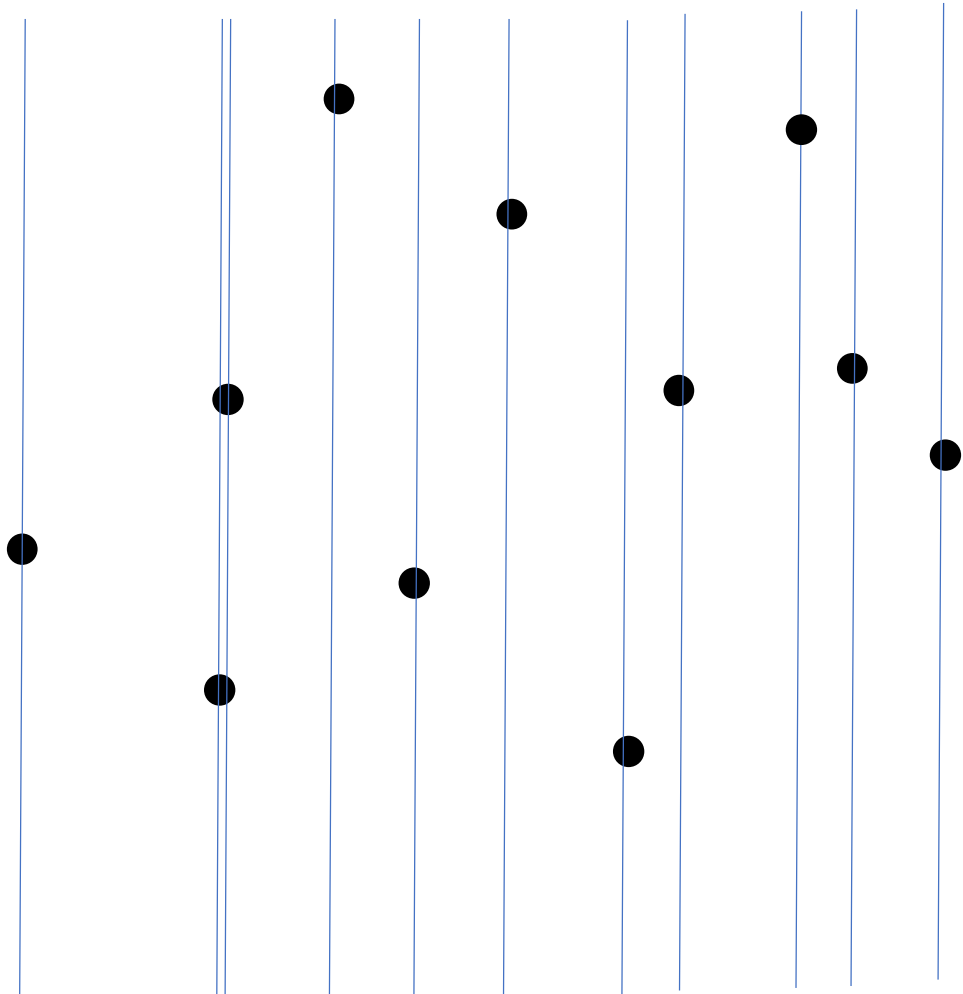


# Monotone Chain

판서로 내용 보강하겠습니다!

좌표 순으로 정렬해서  
가장 작은 좌표부터 본다.

$$O(n \lg n)$$



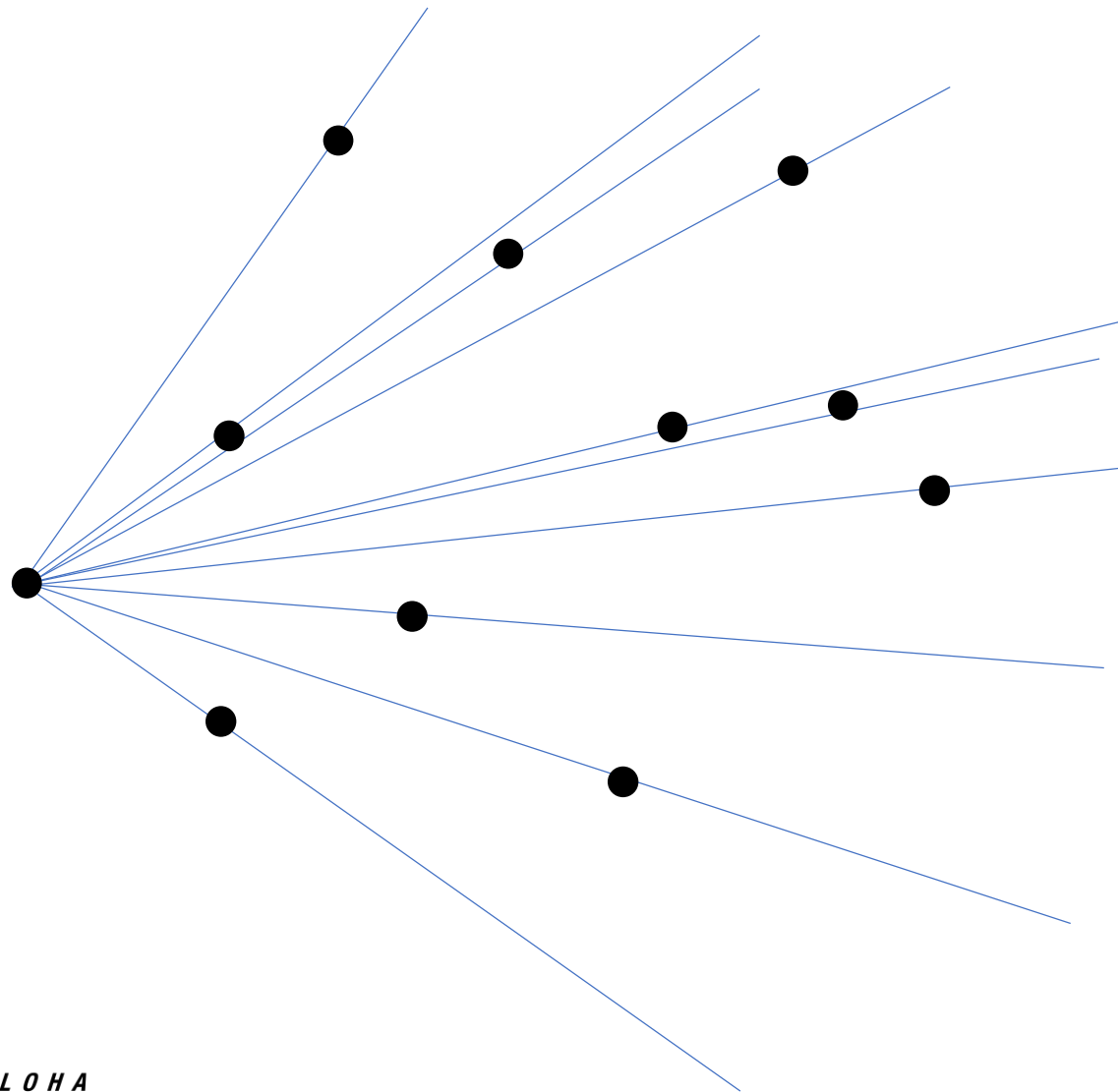


# Graham Scan

판서로 내용 보강하겠습니다!

각도 순으로 정렬해서  
가장 작은 각도부터 본다.

$$O(n \lg n)$$



ALPHA  
The algorithm club.

#2

# 구현

- 그러면 이걸 대체 어떻게 구현할까 -



# CCW

외적을 응용

$$\vec{a} \times \vec{b} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix} = a_x b_y - a_y b_x$$

$$(A_x B_y + B_x C_y + C_x A_y) - (A_y B_x + B_y C_x + C_y A_x)$$

```
long long ccw(const Point& A, const Point& B, const Point& C){  
    return 1LL*(B.x-A.x)*(C.y-A.y) - 1LL*(B.y-A.y)*(C.x-A.x);  
}
```

[출처] [볼록 껍질\(Convex Hull\)](#) (수정: 2019-01-22) | 작성자 [라이](#)

# Pseudocode (Monotone Chain)

x좌표(같으면 y좌표)순으로 정렬한다.  
비어있는 리스트 U와 L을 준비한다.

```
for i = 1, 2, ..., n:  
    while L의 원소가 2개 이상이면서 L의 가장 위와 그 밑에 대해 CCW를 해서 음수이면  
        가장 위의 원소를 삭제한다.  
    P[i]를 L에 추가한다.
```

```
for i = n, n-1, ..., 1:  
    while U의 원소가 두개 이상이면서 U의 가장 위와 그 밑에 대해 CCW를 해서 음수이면  
        가장 위의 원소를 삭제한다.  
    P[i]를 U에 추가한다.
```

각 리스트에서 마지막 원소를 지운다(왜냐하면 다른 리스트의 첫 원소이기 때문이다).  
두 리스트를 합친다.

# Pseudocode (Graham Scan)

x좌표가 가장 작은 점에 대해 기울기순으로 정렬한다.  
비어있는 리스트 C를 준비한다.

```
for i = 1, 2, ..., n:  
    while L의 원소가 2개 이상이면서 C의 가장 위와 그 밑에 대해 CCW를 해서 음수이면  
        가장 위의 원소를 삭제한다.  
    P[i]를 L에 추가한다.
```

# 스택 사용하는 법

```
stack<Point> st;
st.push( points[ 0 ] );
st.push( points[ 1 ] );
for ( int i=2; i != n; i++ ) {
    Point &nowP = points[ i ];
    Point top, semiTop;
    do{
        top = st.top();
        st.pop();
        if( st.empty() ){ break; }
        semiTop = st.top();
    }while( ccw( semiTop, top, nowP ) <= 0 );
    st.push( top );
    st.push( nowP );
}
```

Point의 배열 points가 있을 때  
첫번째와 두번째 원소를 스택에 넣고 시작한다.

위의 두개를 뽑아서 현재 점(nowP)과 CCW하고  
조건에 만족하지 않으면 스택을 pop한다.

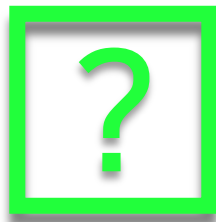
# 연습 문제



BOJ 10254  
고속도로



BOJ 2254  
감옥 건설



BOJ 1708  
블록껍질



BOJ 7420  
맹독 방벽



BOJ 3878  
점 분리



BOJ 9240  
로버트 후드



BOJ 6850  
Cows



# Conclusion

CCW(CounterClockWise)라는 함수를 이용하여 볼록껍질을 구해보았습니다.  
더 적은 시간 복잡도로 순서를 \*만들어서\* 시간 복잡도를 줄일 수 있다는 것을 알 수 있었습니다.

발표에 대해 조언해주신 성창호 선배님께 감사드립니다.



# 수고 많으셨습니다!



ALPHA  
The algorithm club.