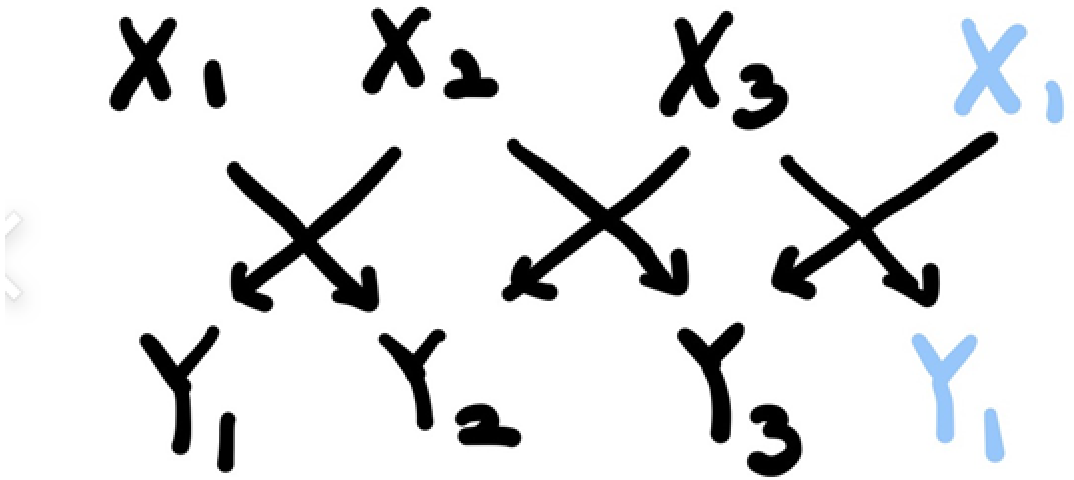
CCW , 교점 판별 원리 및 구현

2019575047 이지민

CCW

- Ccw 는 평면상의 3개의 점과 관련된 점들의 위치 관계를 판단하는 알고리즘
- A(x1,y1) B(x2,y2) C(x3,y3) 일때,

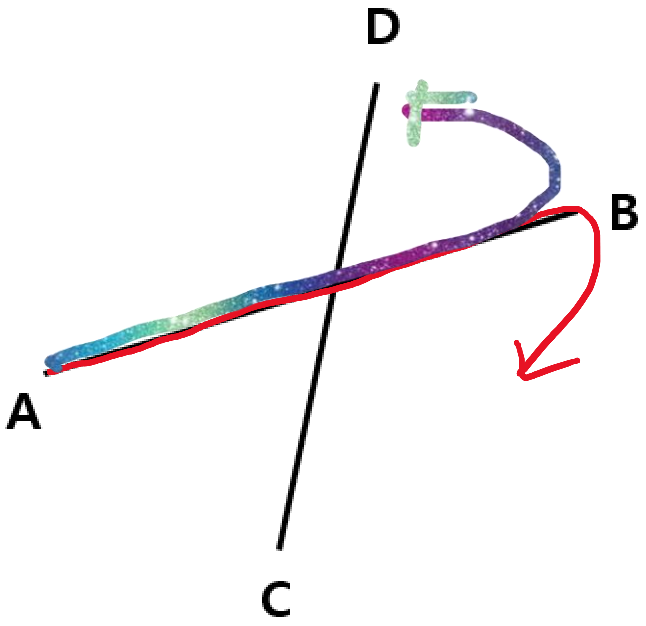
CCW = (x1y2 + x2y3 +x3y1) – (x2y1 +x3y2 +x1y3)



CCW 결과 <0 | CCW 결과==0 | CCW 결과 >0

```java
private static void ccw(int x1, int x2, int x3, int y1, int y2,int y3) {
    // TODO Auto-generated method stub
    int a = x1*y2 + x2*y3 + x3*y1;
    int b = x2*y1 + x3*y2 + x1*y3;
    if (a-b<0) {
        System.out.println("-1");
    }
    else if (a-b==0) {
        System.out.println("0");
    }
    else {
        System.out.println("1");
    }
}
```

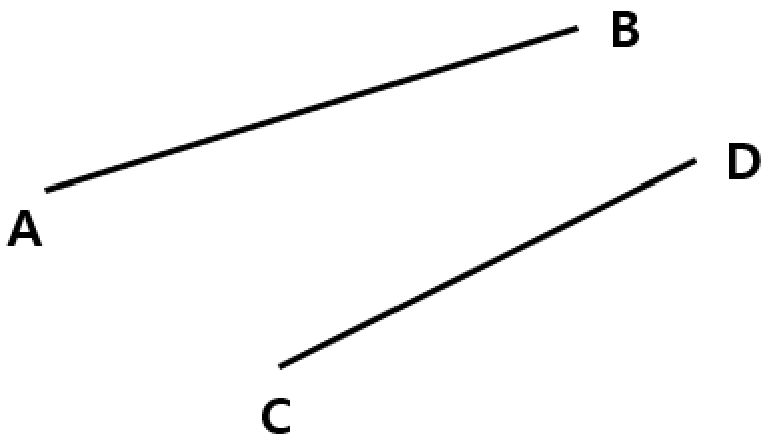
두 선분의 교차 여부



그림과 같이 선분이 교차하게 된다면 선분 AB를기준으로
나머지 한 선분의 두 꼭짓점이 서로 다른 방향에 위치하게 된다.
즉 , C 와 D가 선분 AB 를 기준으로 반대 방향에 위치하게 됩니다.
A , B , C 는 시계 방향의 방향 관계를 가지고,
A, B ,D 는 시계 반대 방향의 관계를 가지는 것을 알 수 있음

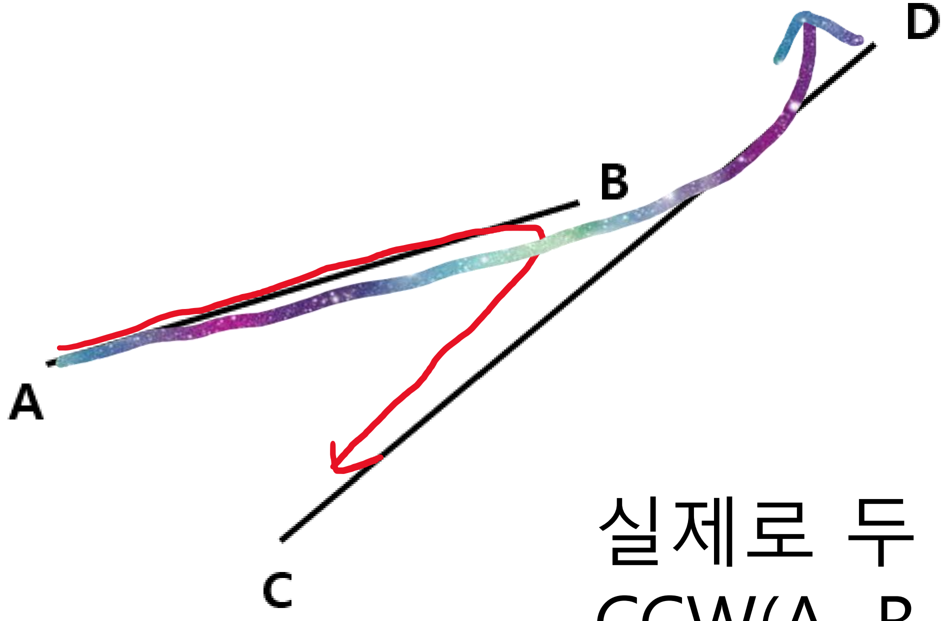
CCW(A,B,C) x CCW(A,B,D) < 0 을 만족하면 교차

교차 하지 않는 경우



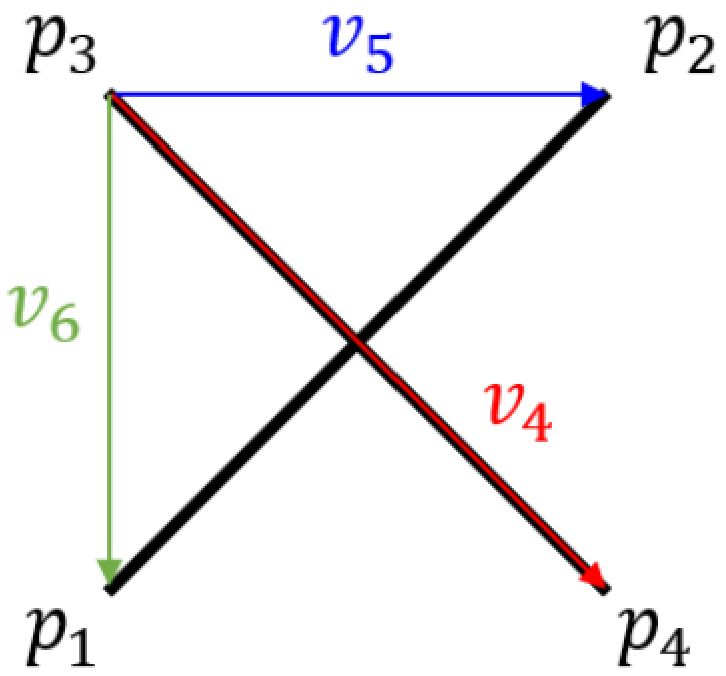
CCW(A, B, C) x CCW(A, B, D) >0 인 경우

예외



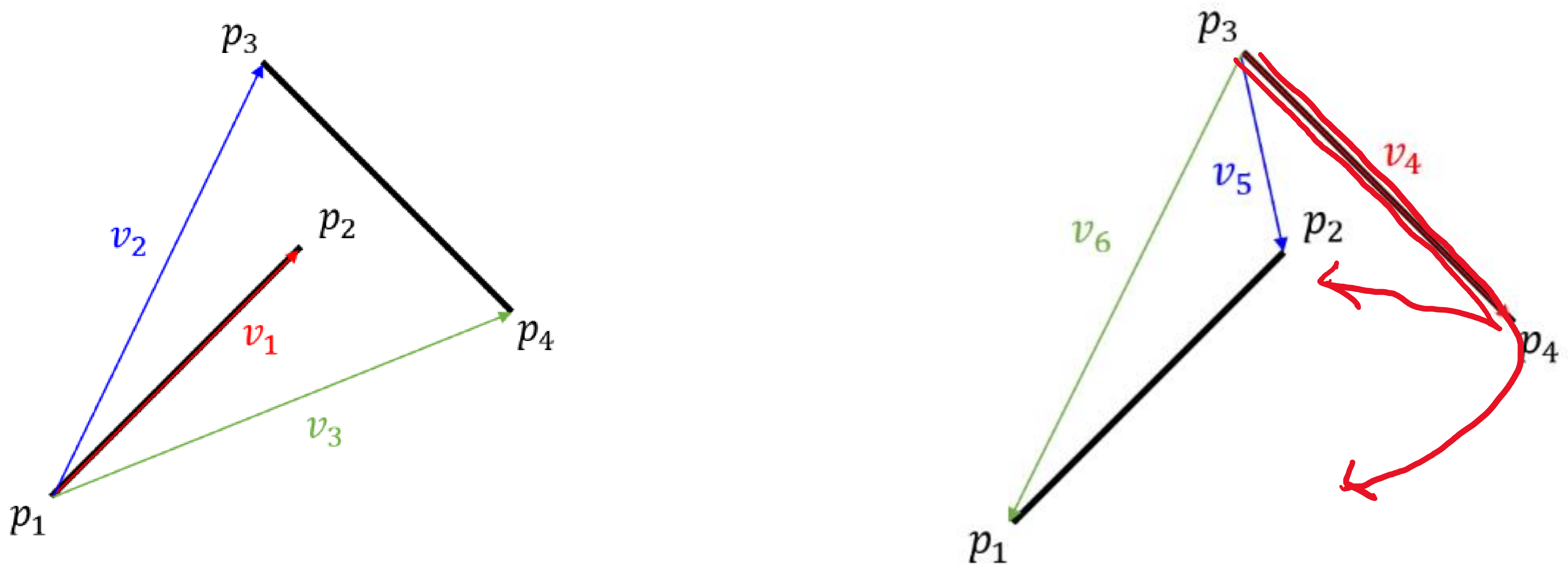
실제로 두 선분이 만나지 않는 경우이지만
CCW(A, B, C) x CCW(A, B, D) <0 을 만족하는
경우가 존재 할 수 있다

해결방안

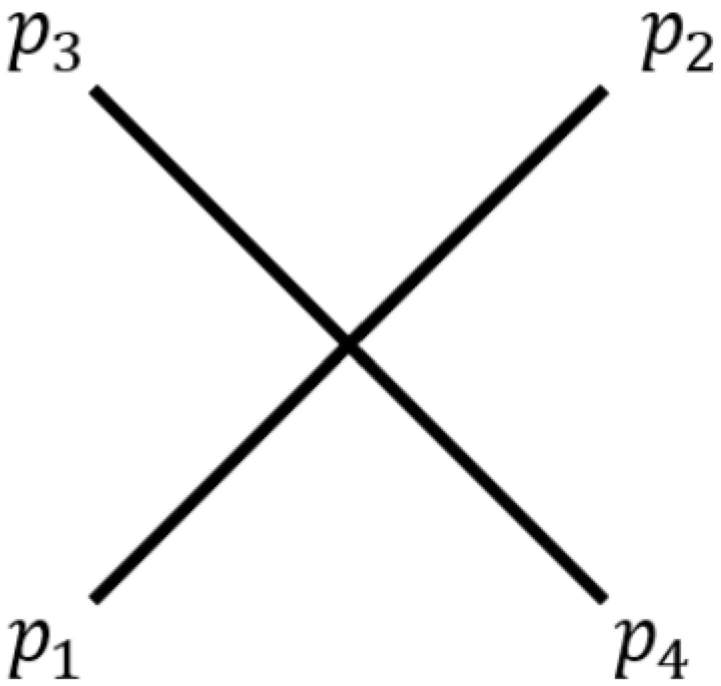


벡터들의 방향성을 확인하면 된다.
V4 = p4 - p3
V5 = p2 – p3
V6 = p1 – p3 일때,
V4 -> v5 로 반시계 방향을 띄고,
V4 -> v6 로 시계 방향을 띄어서
벡터들의 방향성이 다름을 통하여 선분이 교차
한다고 말 할 수 있음

앞의 예외 해결



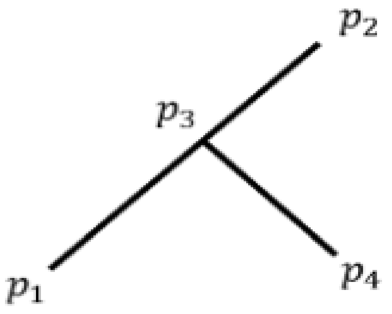
정리

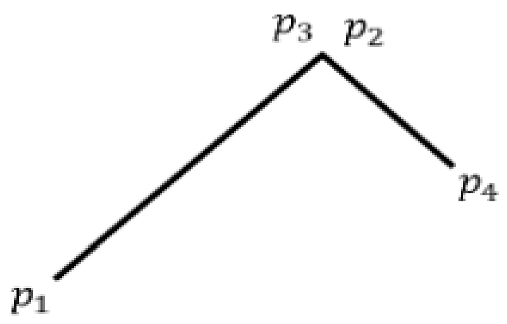


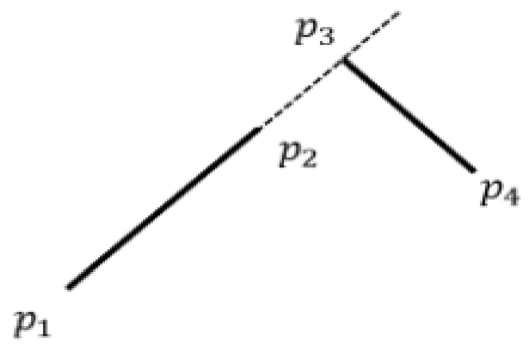
p1 – p2 – p3 의 방향과 p1 – p2 – p4 의 방향이 반대 이고,
P3 –p4 – p1 의 방향과 p3 – p4 – p2 의 방향도 반대

함수로 나타내면
CCW (p1, p2, p3) * CCW(p1, p2, p4)< 0 &
CCW (p3, p4, p1) * CCW (p3, p4, p2) < 0
이여야 함

직선에 있는 경우







CCW (p1, p2, p3) * CCW (p1, p2, p4) ==0 만족하거나
CCW (p3, p4, p1) * CCW (p3, p4, p22) ==0 인 경우
(둘 중 하나만 0인경우)

출처

- 책 : do it 알고리즘 코딩 테스트
- https://rebro.kr/12
- https://gaussian37.github.io/math-algorithm-line_intersection/