Algorithm/Algorithm Study

CCW(CounterClockWise) 알고리즘

code0xff 2017. 11. 21. 16:32

CCW(CounterClockWise)는 외적을 이용해서 두 벡터의 상대적 위치를 파악하는 알고리즘이다. 좀더 구체적으로 설명하자면 하나의 정점을 기준으로 다른 두 정점으로 향하는 벡터가 존재할 때 각 정점의 상대적 위치를 판단하는 공식이다. 만일 하나의 정점에서 다른 정점으로 향하는 벡터를 기준으로 왼쪽에 존재하는 정점의 경우 외적은 양의 값을 갖게 되고, 상대적으로 우측에 있게 되면 외적은 음의 값을 갖게된다. 마지막으로 세 점이 하나의 직선 위에 놓이게 되었을 때 외적은 0이 된다. 이를 이용하여 다각형의 넓이 구하기, 정점의 위치판단(다각형 내부/외부 위치 구분), 볼록다각형 그리기(Convex Hull) 등 다양한 기하 문제를 풀어낼 수 있다.

2차원 평면상에 각각 x, y 좌표를 갖고 있는 세점 (p1, p2, p3)가 존재한다고 했을 때, CCW 공식은 아래와 같다.
s = (p1.x * p2.y + p2.x * p3.y + p3.x * p1.y) - (p1.y * p2.x + p2.y * p3.x + p3.y * p1.x)

만일 결과값인 s가 0보다 작다면(s < 0) p1, p2 벡터를 기준으로 했을 때 p3는 오른쪽에 존재하는 정점이 되고,
s가 0보다 크다면(s > 0) p1, p2 벡터를 기준으로 했을 때 p3는 왼쪽쪽에 존재하는 정점이 된다.
마지막으로 s가 0이라면(s == 0) 세 정점은 한 직선 위에 존재하게 된다.

실제 문제에서는 어떻게 구현되는지 살펴보자.

11758번: CCW - Baekjoon Online Judge
https://www.acmicpc.net/problem/11758

제목부터 정직한 CCW 문제다. 이 문제는 세 정점을 주고 순서대로 이었을 때 어떤 방향을 이루고 있는지 판단하는 문제이다. 여기서는 어떠한 방향을 이루고 있는지만 구하면 되기 때문에 실제 외적을 구할 필요는 없고 결과값이 아닌 양수인지, 음수인지, 0 인지 부호를 판단만 해주면 된다.

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.StringTokenizer;

class Main {
    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st = null;

        st = new StringTokenizer(br.readLine().trim());
        Point p1 = new Point(Long.parseLong(st.nextToken()), Long.parseLong(st.nextToken()));
        st = new StringTokenizer(br.readLine().trim());
        Point p2 = new Point(Long.parseLong(st.nextToken()), Long.parseLong(st.nextToken()));
        st = new StringTokenizer(br.readLine().trim());
        Point p3 = new Point(Long.parseLong(st.nextToken()), Long.parseLong(st.nextToken()));

        System.out.println(ccw(p1, p2, p3));
    }

    static int ccw(Point p1, Point p2, Point p3) {
        long tmp = p1.x * p2.y + p2.x * p3.y + p3.x * p1.y;
        tmp -= (p1.y * p2.x + p2.y * p3.x + p3.y * p1.x);

        if (tmp < 0)
            return -1;
        if (tmp > 0)
            return 1;
        return 0;
    }
}

class Point {
    long x;
    long y;

    Point(long x, long y) {
        this.x = x;
        this.y = y;
    }
}
```


2166번: 다각형의 면적 - Baekjoon Online Judge
https://www.acmicpc.net/problem/2166

다각형의 면적은 CCW를 이용해서 벡터의 외적을 이용하여 주어진 다각형의 넓이를 구하는 문제이다. 문제에서는 친절하게도 다각형을 그리는 순서대로 점의 좌표가 주어졌기 때문에 배열이나 리스트에 순서대로 점들의 위치정보를 저장해뒀다가 세 점씩 묶어서 외적을 구해주고 이 값들을 전부 더해주면 된다.

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.StringTokenizer;

class Main {
    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        StringTokenizer st = null;

        int N = Integer.parseInt(br.readLine().trim());

        Point[] point = new Point[N + 1];

        for (int i = 1; i <= N; i++) {
            st = new StringTokenizer(br.readLine().trim());

            long x = Long.parseLong(st.nextToken());
            long y = Long.parseLong(st.nextToken());

            point[i] = new Point(x, y);
        }

        long ans = 0;
        for (int i = 2; i < N; i++) {
            ans += ccw(point[1], point[i], point[i + 1]);
        }
        ans = Math.abs(ans);
        if (ans % 2 == 1)
            System.out.println((ans / 2) + ".5");
        else
            System.out.println((ans / 2) + ".0");
    }

    static long ccw(Point p1, Point p2, Point p3) {
        long tmp = p1.x * p2.y + p2.x * p3.y + p3.x * p1.y;
        tmp -= (p1.y * p2.x + p2.y * p3.x + p3.y * p1.x);

        return tmp;
    }
}

class Point {
    long x;
    long y;

    Point(long x, long y) {
        this.x = x;
        this.y = y;
    }
}
```


'Algorithm > Algorithm Study' 카테고리의 다른 글

태그 #Algorithm, #CCW, #CounterClockWise, #다각형의 넓이, #알고리즘

'Algorithm/Algorithm Study' Related Articles

NO IMAGE
1654번: 랜선 자르기 - Baekjoon Online Judge

NO IMAGE
2162번: 선분 그룹 - Baekjoon Online Judge

NO IMAGE
사이클 찾기: DFS

NO IMAGE
강한 연결 요소 (SCC: Strongly Connected...