

# Point in Triangle

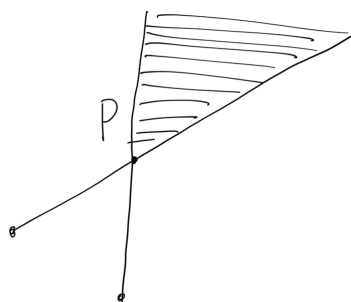
ALPS 2020년 7월 내부대회 N번

출제자: 노세운

Constraints	Time Complexity
$N \leq 300000$	$O(N \cdot \log N)$

## 풀이

삼각형의 세 점 중 두 점이 결정되었을 때 나머지 한 점은 다음 그림과 같이 두 점을  $P$ 와 이은 직선 사이의 영역에 존재해야 합니다.



점들을  $P$ 를 기준으로 하여 반시계 혹은 시계방향으로 정렬한 뒤 적당히 prefix sum이나 스위핑을 이용하면 두 점이 결정되었을 때 빗금친 영역 안의 점의 개수를 구할 수 있습니다. 따라서 간단히  $O(N^2)$  풀이를 얻을 수 있습니다.

아직은 부족합니다. 한 점을 고정시키고 나머지 한 점을 이동시키면서 관찰해 봅시다.

점  $a[i]$ 와  $a[j]$ 를 골랐을 때 빗금친 영역 안의 점의 개수는 prefix sum을 이용하여  $c[j] - b[i]$  꼴로 표현할 수 있습니다.

따라서 점  $a[i]$ 를 한 꼭지점으로 하는  $P$ 를 포함하는 삼각형의 개수는  $\sum_j^{ccw(a[i], P, a[j]) < 0} c[j] - b[i]$ 입니다.

고정된  $i$ 에 대해 가능한  $j$ 는 정렬했을 때 연속적으로 분포하므로 위 식은 prefix sum과 스위핑을 이용하여  $O(1)$ 에 계산 가능합니다.

삼각형을 중복해서 세지 않으려면  $a[i]$ 의 x좌표가  $P$ 보다 작은 것만 세도록 하면 됩니다.

모든 삼각형의 개수는  $O(N)$ 에 계산할 수 있으며 정렬을 해야하기 때문에 총 시간복잡도는  $O(N \cdot \log N)$ 입니다. 세 점이 같은 직선 위에 있을 수 있기 때문에 구현이 까다롭습니다.

## 정해 코드

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
typedef pair<ll, ll> pp;
#define x first
```

```

#define y second
int ccw(pp a,pp b,pp c){
    ll k=a.x*b.y+b.x*c.y+c.x*a.y-a.y*b.x-b.y*c.x-c.y*a.x;
    if(k==0)return 0;
    if(k<0)return -1;
    return 1;
}
const int MX=3e5+5;
pp p;
pp a[MX];
ll s[MX];
ll s2[MX];
ll ss[MX];
int cr[MX];
int n;

int main(){
    scanf("%d",&n);
    int i,j;
    scanf("%lld %lld",&p.x,&p.y);
    for(i=1 ; i<=n ; i++)scanf("%lld %lld",&a[i].x,&a[i].y);

    sort(a+1,a+n+1,[&](pp q,pp w){
        if(q.x>=p.x && w.x<p.x)return false;
        if(q.x<p.x && w.x>=p.x)return true;
        ll k=ccw(q,p,w);
        return k<0;
    });

    int t=0;
    for(i=1 ; i<=n ; i++){
        if(a[i].x>p.x){
            t=i;
            break;
        }
    }

    if(t==0 || t==1){
        printf("0");
        return 0;
    }

    int k=1;
    for(i=1 ; i<t ; i++){
        while(k<t && ccw(a[i],p,a[k])==0 )k++;
        cr[i]=k-1;
    }

    int jj=j=t;

    for(i=1 ; i<t ; i++){
        j=max(j,cr[i]+1);
        jj=max(jj,cr[i]+1);
        while(j<=n && ccw(a[i],p,a[j])<0)j++;
        while(jj<=n && ccw(a[i],p,a[jj])<=0)jj++;
        s2[i]=jj-1;
        s[i]=j-1;
    }
}

```

```
}

for(i=t ; i<=n ; i++)s[i]=n;
for(i=1 ; i<=n ; i++)ss[i]=ss[i-1]+s[i];

long long ans=0;
for(i=1 ; i<t ; i++){
    long long cnt=s[i]-cr[i];
    long long S=ss[s[i]]-ss[cr[i]];
    ans+=S-cnt*s2[i];
}
cout<<ans;
return 0;
}
```