Crack – a – Hack

# Triangles containing the origin

Course: ALGORITHMIC PROBLEM SOLVING

Course Code: 17ECSE309

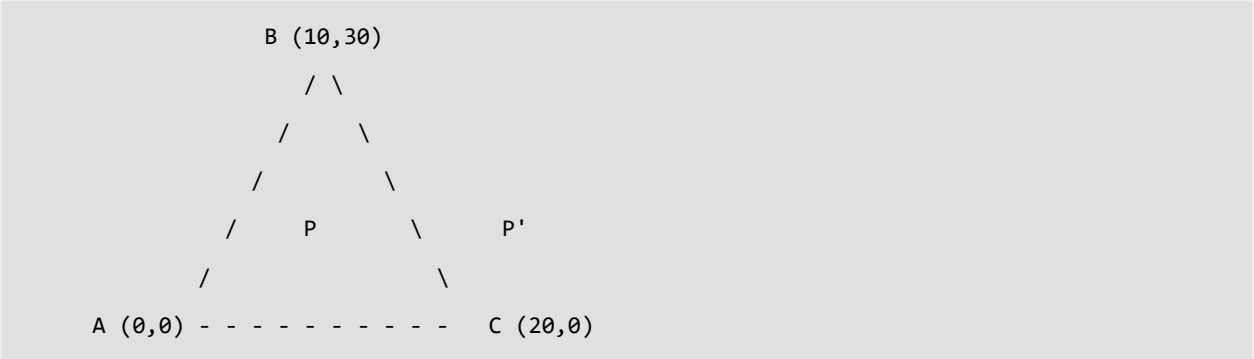
By: Pavankumar Hittalamani

01FE15BCS129

## Introduction:

Given three corner points of a triangle, and one more point P. Write a function to check whether P lies within the triangle or not using area method.

## Example:




For example, consider the following program, the function should return true for P(10, 15) and false for P'(30, 15)

Let the coordinates of three corners be (x1, y1), (x2, y2) and (x3, y3). And coordinates of the given point P be (x, y)

1) Calculate area of the given triangle, i.e., area of the triangle ABC in the above diagram. Area A = [ x1(y2 – y3) + x2(y3 – y1) + x3(y1-y2)]/2

2) Calculate area of the triangle PAB. We can use the same formula for this. Let this area be A1.

3) Calculate area of the triangle PBC. Let this area be A2.

4) Calculate area of the triangle PAC. Let this area be A3.

5) If P lies inside the triangle, then A1 + A2 + A3 must be equal to A.

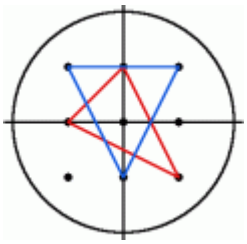
## Hackerrank Question Link:

Project Euler 184 : Triangles containing the origin

## Problem Description:

Consider the set $I_r$ of points (x, y) with integer co-ordinates in the interior of the circle with radius r, centered at the origin, i.e. $x^2 + y^2 < r^2$.

For a radius of 2, $I_2$ contains the nine points (0, 0), (1, 0), (1, 1), (0, 1), (-1, 1), (-1, 0), (-1, -1), (0, -1) and (1, -1). There are eight triangles having all three vertices in $I_2$ which contain the origin in the interior. Two of them are shown below, the others are obtained from these by rotation.



For a radius of 3, there are 360 triangles containing the origin in the interior and having all vertices in $I_3$ and for $I_5$ the number is 10600.

How many triangles are there containing the origin in the interior and having all three vertices in $I_r$?

**Constraints:** $2 < r < 10^6$

## Solution:

The solution is written in C++:

## Functions:

Area (a [], b [], c [])

Distance (a [], b [])

```cpp
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>

using namespace std;

int area(int a[], int b[], int c[]) {
    return abs((a[0] - c[0])*(b[1] -a[1]) -
                    (a[0]-b[0])*(c[1]-a[1]));
}

int distance(int a[], int b[]) {
    return (int)abs(sqrt(pow(a[0] - b[0], 2) +
                pow(a[1] - b[1], 2) * 1.0));
}

int main() {

    int A[2];
    int B[2];
    int C[2];
    int P[2];
    P[0]=0;
    P[1]=0;

    int n,z=0;
    cin >> n;
    int result = 0;
    n = n - 1 ;
    int i,j,k,l,x,y;
    for( i = -n ; i <= n ; i++  ){
        for( j = -n ; j <= n ; j++ ){
            for( k = -n ; k <= n ; k++  ){
                /*if( k == i ){
                    l = j;
                } else {
                    l = -n;
                }*/
                for(l = -n ; l <= n ; l++ ){

                    for( x = -n ; x <= n ; x++  ){
                        /*if( x == k ){
                            y = l;
                        } else {
                            y = -n;
                        }*/
                        for( y = -n; y <= n; y++ ){
                            if( k == x && l == y){
                                continue;
                            }
                            if( i == x && j == y){
                                continue;
                            }
                            if(i == k && j == l){
                                continue;
                            }
                            A[0] = i;
                            A[1] = j;
                            B[0] = k;
                            B[1] = l;
                            C[0] = x;
                            C[1] = y;
                            if(distance(A,P)>= n+1 || distance(B,P) >= n+1 || distance(C,P) >= n+1){
                                continue;
                            }
```

```
                        if(area(A, B, C) == 0 ||
                            area(A, B, P) == 0 ||
                            area(A, P, C) == 0 ||
                            area(P, B, C) == 0){
                            continue;
                        }

                        if (area(A, B, C) ==
                            area(A, B, P) +
                            area(A, P, C) +
                            area(P, B, C))
                            result++;
                    }
                }
            }
        }
    }
    cout << result/6;
    return 0;
}
```

## Time Complexity:

It takes a time complexity of O(n^3).

Optimized using functions to get it reduced by operating after checking validity.

## References:

https://www.geeksforgeeks.org/check-whether-a-given-point-lies-inside-a-triangle-or-not/