

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ТРИЪГЪЛНИЦИ

За решаване на задачата ще използваме следното достатъчно условие за пресичане на два триъгълника:

Два триъгълника имат поне една обща точка, ако:

- връх на единия триъгълник лежи в другия триъгълник или
- нито един връх от единия триъгълник не лежи в другия триъгълник, но страна от единия и страна от другия триъгълник имат обща точка.

За реализиране на програмата дефинираме структура `Point`

```
struct Point
{
    int x, y;
};
```

Следващата функция определя ориентацията на ъгъл с връх точка A , първо рамо – лъча AB^{\rightarrow} и второ рамо – лъча AC^{\rightarrow} . Използвани са свойства на кръстосаното произведение на два вектора.

```
int direction(Point A, Point B, Point C)
{
    int a1=B.x-A.x, a2=B.y-A.y;
    int b1=C.x-A.x, b2=C.y-A.y;
    int p=a1*b2, q=a2*b1;
    if(p>q) return +1;
    if(p<q) return -1;
    return 0;
}
```

С помощта на следващите две функции може да проверим дали две отсечки имат обща точка. С функцията `onSegment` проверяваме дали точка C лежи на отсечката AB при условие, че точките A , B и C лежат на една права.

```
bool onSegment(Point A, Point B, Point C)
{
    return min(A.x,B.x)<=C.x && C.x<=max(A.x,B.x) &&
           min(A.y,B.y)<=C.y && C.y<=max(A.y,B.y);
}
```

```
bool intersectSegments(Point A, Point B, Point C, Point D)
{
    int d1 = direction(A,B,C);
    int d2 = direction(A,B,D);
    int d3 = direction(C,D,A);
    int d4 = direction(C,D,B);
    if (d1*d2<0 && d3*d4<0) return true;
    if (d1==0 && onSegment(A,B,C)) return true;
    if (d2==0 && onSegment(A,B,D)) return true;
    if (d3==0 && onSegment(C,D,A)) return true;
    if (d4==0 && onSegment(C,D,B)) return true;
    return false;
}
```

С функцията `onTriangle` проверяваме дали точка M е от триъгълника ABC .

```
bool onTriangle(Point A, Point B, Point C, Point M)
{
    int d1 = direction(A,B,M);
    int d2 = direction(B,C,M);
    int d3 = direction(C,A,M);
    if (d1>=0 && d2>=0 && d3>=0) return true;
    if (d1<=0 && d2<=0 && d3<=0) return true;
    return false;
}
```

Функцията `intersectTriangles` проверява дали триъгълниците ABC и PQR имат обща точка, като използва описаното по-горе достатъчно условие.

```
bool intersectTriangles(Point A, Point B, Point C, Point P,
Point Q, Point R)
{
    if (onTriangle(A,B,C,P) || onTriangle(A,B,C,Q) ||
onTriangle(A,B,C,R)) return true;
    if (onTriangle(P,Q,R,A) || onTriangle(P,Q,R,B) ||
onTriangle(P,Q,R,C)) return true;
    if (intersectSegments(A,B,P,Q) ||
intersectSegments(A,B,P,R) || intersectSegments(A,B,Q,R))
return true;
    if (intersectSegments(B,C,P,Q) ||
intersectSegments(B,C,P,R) || intersectSegments(B,C,Q,R))
return true;
    if (intersectSegments(C,A,P,Q) ||
intersectSegments(C,A,P,R) || intersectSegments(C,A,Q,R))
return true;
    return false;
}
```

Остава последната функция да се използва за всяка от двойките триъгълници ABC и $P_1Q_1R_1$, ABC и $P_2Q_2R_2$, ABC и $P_3Q_3R_3$, ABC и $P_4Q_4R_4$.

Автор: Младен Манев