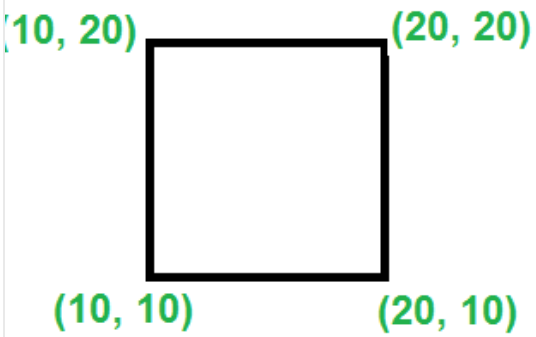
GeeksQuiz

Computer science mock tests for geeks

How to check if given four points form a square

Given coordinates of four points in a plane, find if the four points form a square or not.
To check for square, we need to check for following.
a) All fours sides formed by points are same.
b) The angle between any two sides is 90 degree. (This condition is required as Quadrilateral also has same sides.



We strongly recommend to minimize your browser and try this yourself first.

The idea is to pick any point and calculate its distance from rest of the points. Let the picked picked point be 'p'. To form a square, distance of two points must be same from 'p', let this distance be d. The distance from one point must be different from that d and must be equal to √2 times d. Let this point with different distance be 'q'.
The above condition is not good enough as the the point with different distance can be on the other side. We also need to check that q is at same distance from 2 other points and this distance is same as d.

Below is C++ implementation of above idea.

```
// A C++ program to check if four given points form a square or not.
#include<iostream>
using namespace std;

// Structure of a point in 2D space
```

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

// A utility function to find square of distance
// from point 'p' to poitn 'q'
int distSq(Point p, Point q)
{
    return (p.x - q.x)*(p.x - q.x) +
           (p.y - q.y)*(p.y - q.y);
}

// This function returns true if (p1, p2, p3, p4) form a
// square, otherwise false
bool isSquare(Point p1, Point p2, Point p3, Point p4)
{
    int d2 = distSq(p1, p2);  // from p1 to p2
    int d3 = distSq(p1, p3);  // from p1 to p3
    int d4 = distSq(p1, p4);  // from p1 to p4

    // If lengths if (p1, p2) and (p1, p3) are same, then
    // following conditions must met to form a square.
    // 1) Square of length of (p1, p4) is same as twice
    //    the square of (p1, p2)
    // 2) p4 is at same distance from p2 and p3
    if (d2 == d3 && 2*d2 == d4)
    {
        int d = distSq(p2, p4);
        return (d == distSq(p3, p4) && d == d2);
    }

    // The below two cases are similar to above case
    if (d3 == d4 && 2*d3 == d2)
    {
        int d = distSq(p2, p3);
        return (d == distSq(p2, p4) && d == d3);
    }
    if (d2 == d4 && 2*d2 == d3)
    {
        int d = distSq(p2, p3);
        return (d == distSq(p3, p4) && d == d2);
    }

    return false;
}

// Driver program to test above function
int main()
{
    Point p1 = {20, 10}, p2 = {10, 20},
          p3 = {20, 20}, p4 = {10, 10};
    isSquare(p1, p2, p3, p4)? cout << "Yes": cout << "No";
    return 0;
}
```

Output:

```
Yes
```

This article is contributed by **Anuj**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Category: Algorithms