

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1Solution 2Solution 3

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 #include <unordered_map>
5 using namespace std;
6
7 struct Point {
8     int x;
9     int y;
10 };
11
12 unordered_map<string, bool> getCoordsTable(vector<Point> coords);
13 int getRectangleCount(vector<Point> coords,
14                        unordered_map<string, bool> coordsTable);
15 bool isInUpperRight(Point coord1, Point coord2);
16 string coordToString(Point coord);
17
18 // O(n^2) time | O(n) space - where n is the number of coordinates
19 int rectangleMania(vector<Point> coords) {
20     unordered_map<string, bool> coordsTable = getCoordsTable(coords);
21     return getRectangleCount(coords, coordsTable);
22 }
23
24 unordered_map<string, bool> getCoordsTable(vector<Point> coords) {
25     unordered_map<string, bool> coordsTable;
26     for (Point coord : coords) {
27         string coordString = coordToString(coord);
28         coordsTable.insert({coordString, true});
29     }
30     return coordsTable;
31 }
32
33 int getRectangleCount(vector<Point> coords,
```

Solution 1Solution 2Solution 3

```
1 #include <vector>
2
3 using namespace std;
4
5 struct Point {
6     int x;
7     int y;
8 };
9
10 int rectangleMania(vector<Point> coords) {
11     // Write your code here.
12     return -1;
13 }
14
```

Our Tests

Custom Output

Submit Code

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 #include <unordered_map>
5 using namespace std;
6
7 struct Point {
8     int x;
9     int y;
10 };
11
12 unordered_map<string, bool> getCoordsTable(vector<Point> coords);
13 int getRectangleCount(vector<Point> coords,
14                        unordered_map<string, bool> coordsTable);
15 bool isInUpperRight(Point coord1, Point coord2);
16 string coordToString(Point coord);
17
18 // O(n^2) time | O(n) space - where n is the number of coordinates
19 int rectangleMania(vector<Point> coords) {
20     unordered_map<string, bool> coordsTable = getCoordsTable(coords);
21     return getRectangleCount(coords, coordsTable);
22 }
23
24 unordered_map<string, bool> getCoordsTable(vector<Point> coords) {
25     unordered_map<string, bool> coordsTable;
26     for (Point coord : coords) {
27         string coordString = coordToString(coord);
28         coordsTable.insert({coordString, true});
29     }
30     return coordsTable;
31 }
32
33 int getRectangleCount(vector<Point> coords,
```

```
1 #include <vector>
2
3 using namespace std;
4
5 struct Point {
6     int x;
7     int y;
8 };
9
10 int rectangleMania(vector<Point> coords) {
11     // Write your code here.
12     return -1;
13 }
14
```

