391. Perfect Rectangle

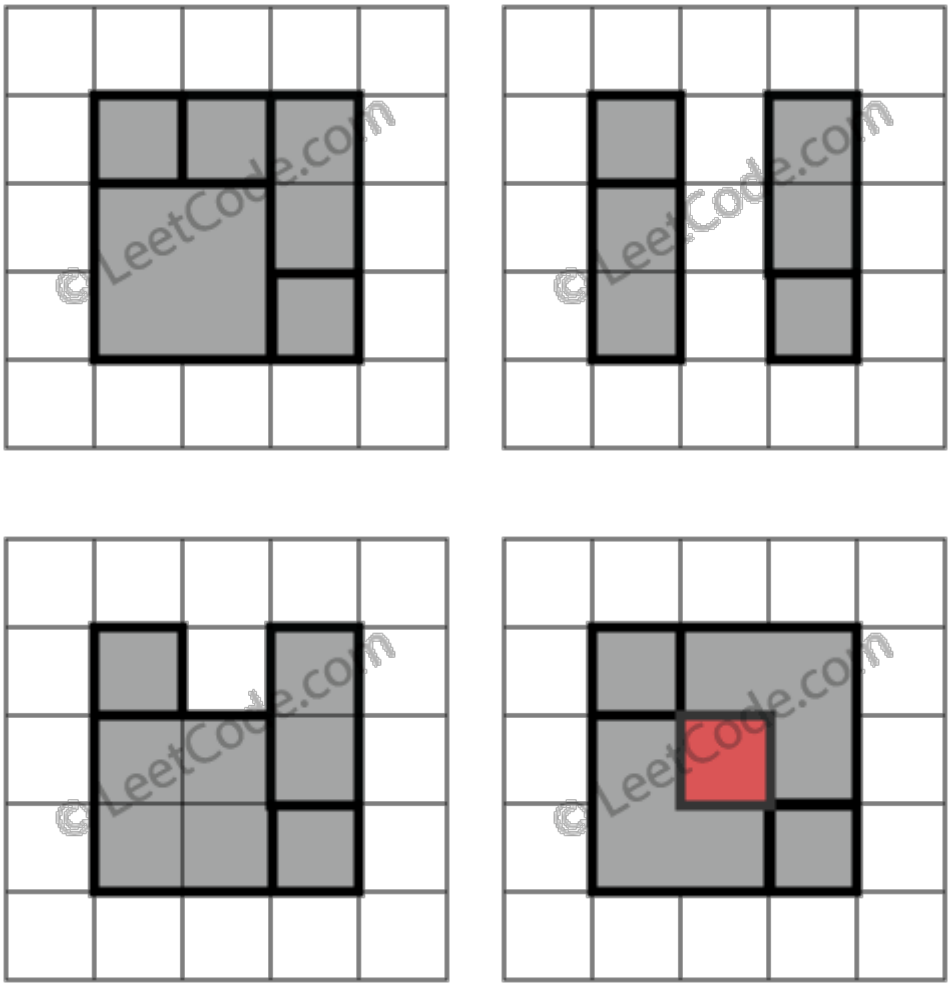
QuestionEditorial Solution

My Submissions

- Total Accepted: **2233**
- Total Submissions: **13165**
- Difficulty: **Hard**

Given N axis-aligned rectangles where N > 0, determine if they all together form an exact cover of a rectangular region.

Each rectangle is represented as a bottom-left point and a top-right point. For example, a unit square is represented as [1,1,2,2]. (coordinate of bottom-left point is (1, 1) and top-right point is (2, 2)).



e. g. 1

```
rectangles = [
  [1,1,3,3],
  [3,1,4,2],
  [3,2,4,4],
  [1,3,2,4],
  [2,3,3,4]
]
```

Return true. All 5 rectangles together form an exact cover of a rectangular region.
e. g. 2
rectangles = [
 [1,1,2,3],
 [1,3,2,4],
 [3,1,4,2],
 [3,2,4,4]
]

Return false. Because there is a gap between the two rectangular regions.
e. g. 3
rectangles = [
 [1,1,3,3],
 [3,1,4,2],
 [1,3,2,4],
 [3,2,4,4]
]

Return false. Because there is a gap in the top center.
e. g. 4
rectangles = [
 [1,1,3,3],
 [3,1,4,2],
 [1,3,2,4],
 [2,2,4,4]
]

Return false. Because two of the rectangles overlap with each other.

```
class Solution {
public:
    bool isRectangleCover(vector<vector<int>>& rectangles) {
        set<string> st;
        int min_x=INT_MAX,min_y=INT_MAX,max_x=INT_MIN,max_y=INT_MIN;
        int area=0;
        for(int i=0;i<rectangles.size();i++)
        {
            vector<int> rect = rectangles[i];
            min_x = min(min_x,rect[0]);
            min_y = min(min_y,rect[1]);
            max_x = max(max_x,rect[2]);
            max_y = max(max_y,rect[3]);
            area+=(rect[2]-rect[0])*(rect[3]-rect[1]);
            string s1 = to_string(rect[0])+"_"+to_string(rect[1]);
            string s2 = to_string(rect[0])+"_"+to_string(rect[3]);
            string s3 = to_string(rect[2])+"_"+to_string(rect[1]);
            string s4 = to_string(rect[2])+"_"+to_string(rect[3]);
            if (st.count(s1)) st.erase(s1); else st.insert(s1);
            if (st.count(s2)) st.erase(s2); else st.insert(s2);
            if (st.count(s3)) st.erase(s3); else st.insert(s3);
            if (st.count(s4)) st.erase(s4); else st.insert(s4);
        }
        string t1 = to_string(min_x)+"_"+to_string(min_y);
        string t2 = to_string(min_x)+"_"+to_string(max_y);
        string t3 = to_string(max_x)+"_"+to_string(min_y);
        string t4 = to_string(max_x)+"_"+to_string(max_y);
        if(!st.count(t1) || !st.count(t2) || !st.count(t3) || !st.count(t4) ||
st.size()!=4) return false;
        return area==(max_x-min_x)*(max_y-min_y);
    }
};
```