Rectangle Area (/problems/rectangle-area/)

Submission Detail

3082 / 3082 test cases passed.

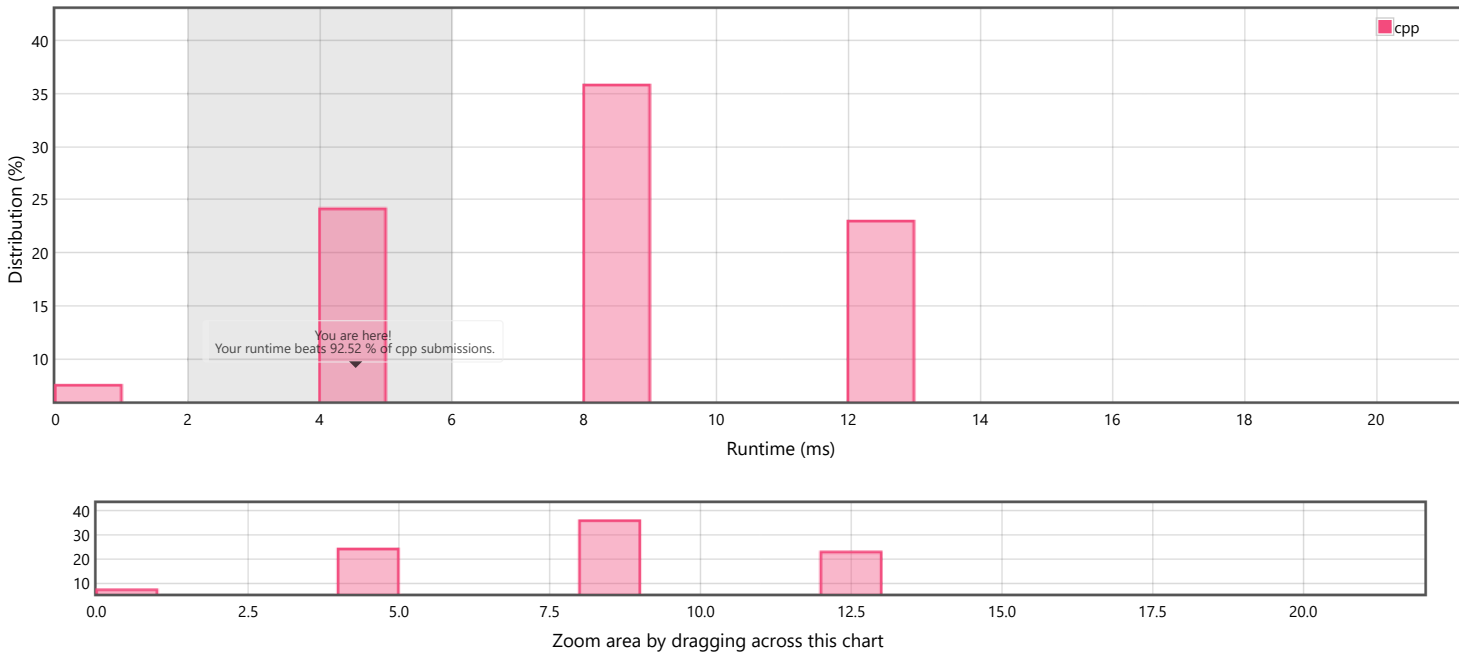
Status: **Accepted**

Runtime: **4 ms**

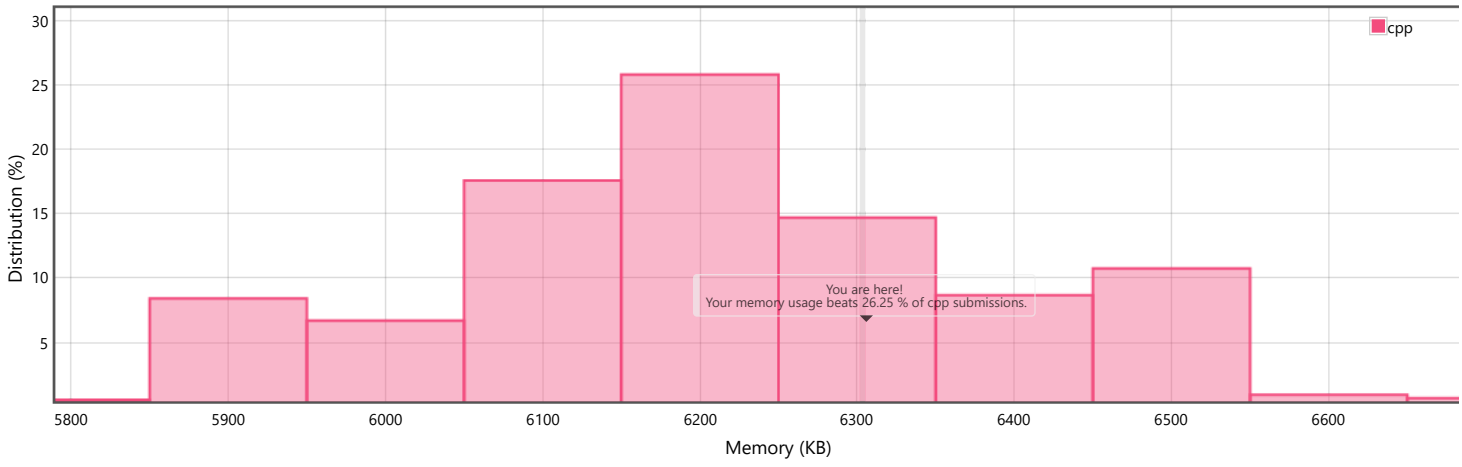
Memory Usage: **6.3 MB**

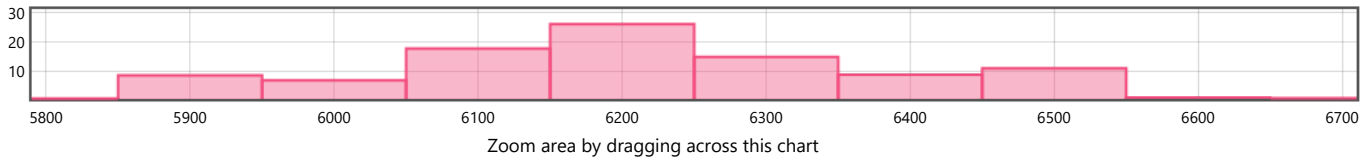
Submitted: **13 hours, 49 minutes ago**

Accepted Solutions Runtime Distribution



Accepted Solutions Memory Distribution





Invite friends to challenge **Rectangle Area**

27

Submitted Code: 13 hours, 49 minutes ago

Language: cpp

Edit Code

```cpp
class Solution {
public:
    int computeArea(int A, int B, int C, int D, int E, int F, int G, int H) {

        int area1=0, area2=0, areaI=0;

        //compute if the given rectangles are FULLY overlaping
        if (A==E && B==F && C==G && D==H)
            {
                area1 = abs((C-A) * (D-B));
                area2 = 0;
            }

        //compute if the given rectangles are not overlapping
        else if (E>=C || A>=G)
            {   //In this case rectangles are not overlaping
                area1 = abs((C-A) * (D-B));
                area2 = abs((G-E) * (H-F));
            }

        else if (F>=D || B>=H)
            {   //In this case rectangles are not overlaping
                area1 = abs((C-A) * (D-B));
                area2 = abs((G-E) * (H-F));
            }
        //following is applicable only if they are overlapping rectangle
        else
            {
                //computing area of the first and second rectangle
                area1 = abs((C-A) * (D-B));
                area2 = abs((G-E) * (H-F));

                //Computing the area of overlapping portion
                int x_dist = abs(min (C,G) - max(A,E));
                int y_dist = abs(min(D,H) - max(B,F));

                areaI = x_dist * y_dist;
            }

        //returning the total area
        return ((area1 + area2)-areaI);

    }
};
```

Back to problem (/problems/rectangle-area/)