Lowest Common Ancestor of a Binary Tree (/problems/lowest-common-ancestor-of-a-binary-tree/)

Submission Detail

31 / 31 test cases passed.

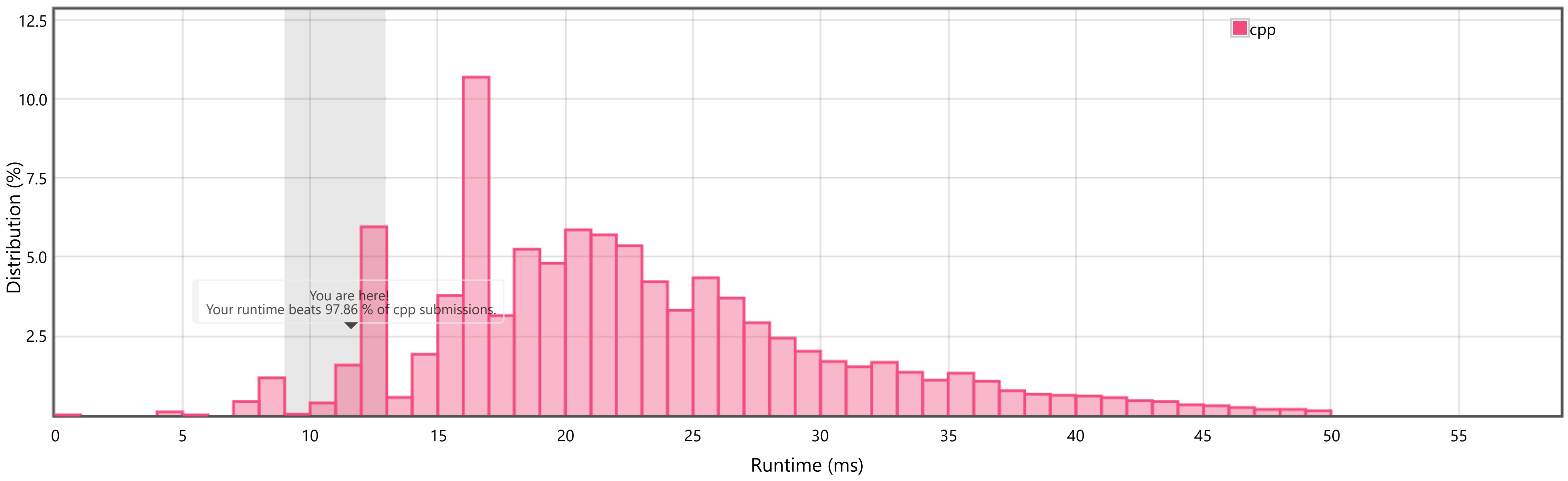
Status: Accepted

Runtime: **11 ms**

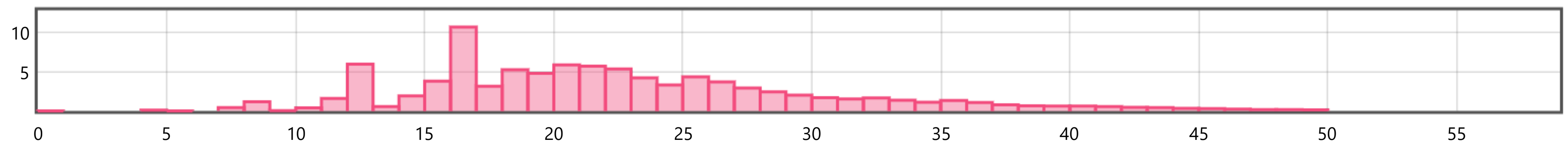
Memory Usage: **17.4 MB**

Submitted: **0 minutes ago**

Accepted Solutions Runtime Distribution

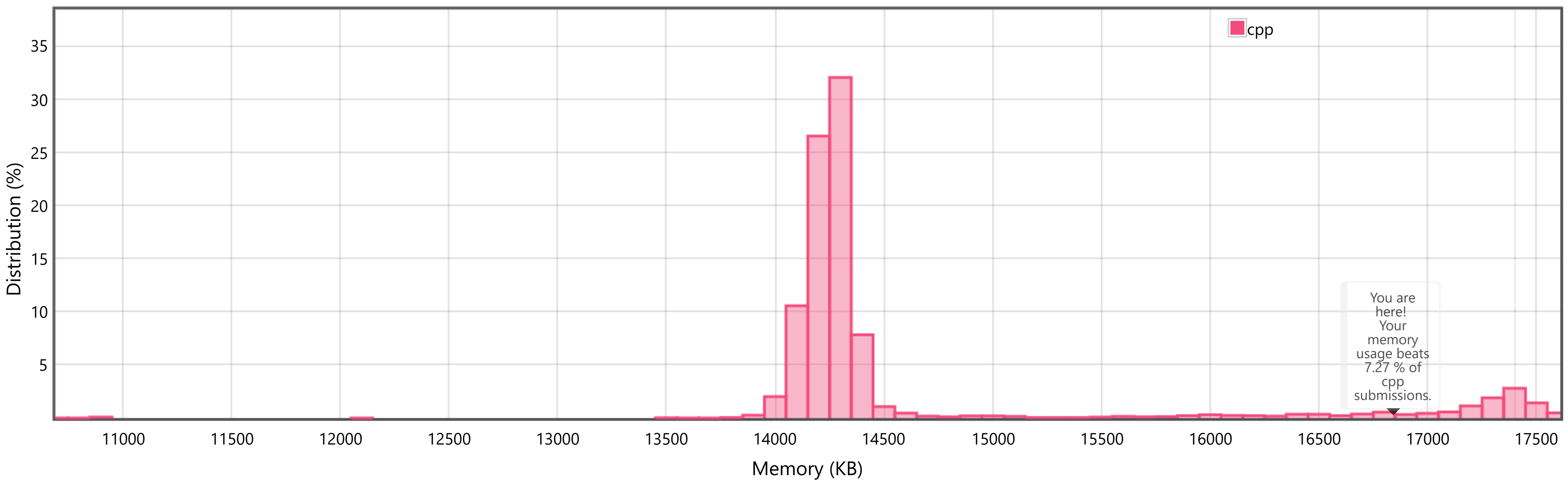


You are here!
Your runtime beats 97.86 % of cpp submissions.

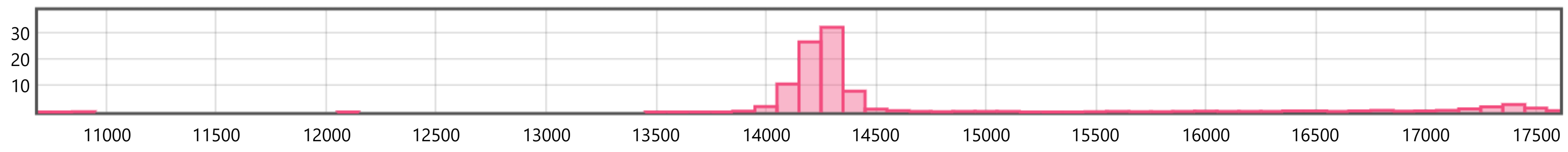


Zoom area by dragging across this chart

Accepted Solutions Memory Distribution



You are here!
Your memory usage beats 7.27 % of cpp submissions.



Zoom area by dragging across this chart

Invite friends to challenge **Lowest Common Ancestor of a Binary Tree**

29

Submitted Code: 0 minutes ago

Language: cpp

Edit Code

```cpp
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class Solution {
public:
    bool solve(TreeNode* root, vector<TreeNode*>& path, TreeNode* p){
        if(root == NULL)
            return false;
        path.push_back(root);
        if(root->val == p->val)
            return true;
        bool l = solve(root->left, path, p);
        bool r = solve(root->right, path, p);
        if(l or r)
            return true;

        path.pop_back();
        return false;

    }
    TreeNode* lowestCommonAncestor(TreeNode* root, TreeNode* p, TreeNode* q) {

        vector<TreeNode*> pathP, pathQ;
        int l = solve(root, pathP, p);
        int r = solve(root, pathQ, q);
        if(!(l and r)) return (new TreeNode(-1));
        int i = 0;
        for(i = 0; i < min(pathP.size(), pathQ.size()); i++){
            if(pathP[i] != pathQ[i])
                return pathP[i - 1];
        }
        if(i == min(pathP.size(), pathQ.size()))
            return pathP[i - 1];

        return (new TreeNode(-1));
    }
};
```

Back to problem (/problems/lowest-common-ancestor-of-a-binary-tree/)

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)