

Pacific Atlantic Water Flow (/problems/pacific-atlantic-water-flow/)

Submission Detail

113 / 113 test cases passed.

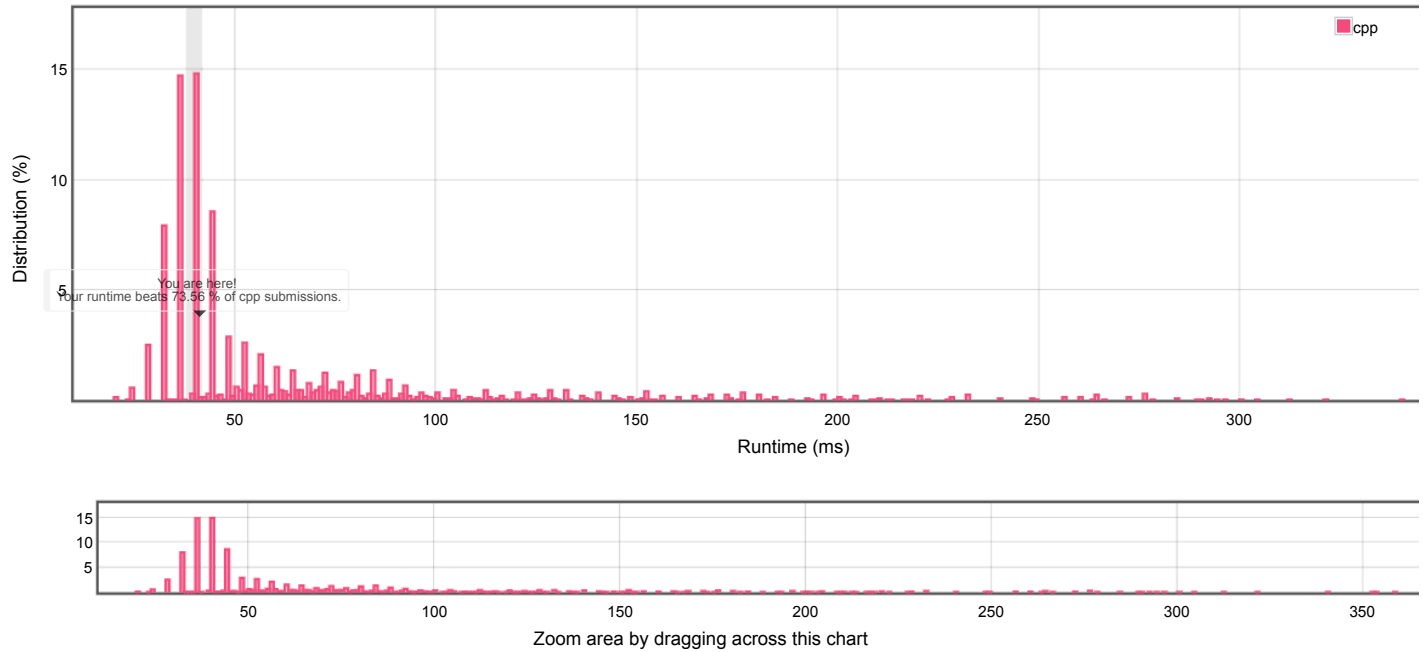
Runtime: **40 ms**

Memory Usage: **17.7 MB**

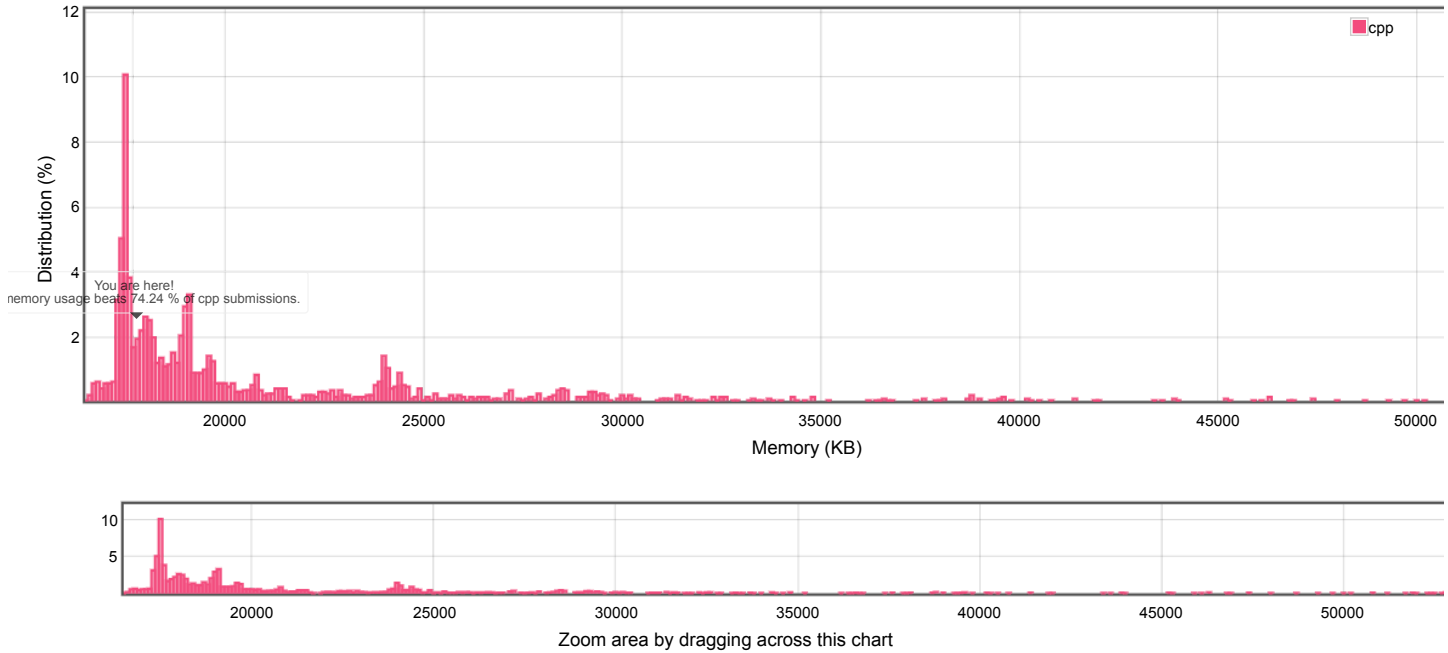
Status: **Accepted**

Submitted: **2 minutes ago**

Accepted Solutions Runtime Distribution



Accepted Solutions Memory Distribution



Invite friends to challenge **Pacific Atlantic Water Flow**

Submitted Code: 2 minutes ago

Language: cpp

Edit Code

```
1 class Solution {
2 public:
3     std::vector<int> direction = {0, 1, 0, -1, 0};
4 }
```

```


5    bool outOfBound(int i, int j, int m, int n) {
6        return (i<0 || j<0 || i>=m || j>=n);
7    }
8
9    void DFS(std::vector<std::vector<int>>& matrix, std::vector<std::vector<bool>>& visited, int i, int j) {
10        int m = matrix.size();
11        int n = matrix[0].size();
12
13        visited[i][j] = true;
14
15        for(int k=1;k<direction.size();k++) {
16            int row = i - direction[k-1];
17            int col = j + direction[k];
18
19            if(!outOfBound(row, col, m, n) && !visited[row][col] && matrix[row][col] >= matrix[i][j] )
20                DFS(matrix, visited, row, col);
21        }
22    }
23    std::vector<std::vector<int>> pacificAtlantic(std::vector<std::vector<int>>& matrix) {
24        if(matrix.size() == 0)
25            return {};
26        int m = matrix.size();
27        int n = matrix[0].size();
28
29        std::vector<std::vector<bool>> pacific(m, std::vector<bool>(n, false));
30        std::vector<std::vector<bool>> atlantic(m, std::vector<bool>(n, false));
31
32        for(int i=0;i<m;i++) {
33            if(!pacific[i][0])
34                DFS(matrix, pacific, i, 0);
35            if(!atlantic[i][n-1])
36                DFS(matrix, atlantic, i, n-1);
37        }
38
39        for(int j=0;j<n;j++) {
40            if(!pacific[0][j])
41                DFS(matrix, pacific, 0, j);
42            if(!atlantic[m-1][j])
43                DFS(matrix, atlantic, m-1, j);
44        }
45
46        std::vector<std::vector<int>> res;
47
48        for(int i=0;i<m;i++) {
49            for(int j=0;j<n;j++) {
50                if(pacific[i][j] && atlantic[i][j])
51                    res.push_back({i, j});
52            }
53        }
54
55        return res;
56    }
57 }

```

[Back to problem \(/problems/pacific-atlantic-water-flow/\)](/problems/pacific-atlantic-water-flow/)

Copyright © 2021 LeetCode

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

 [United States \(/region\)](/region/)