Sliding Puzzle (/problems/sliding-puzzle/)

Submission Detail

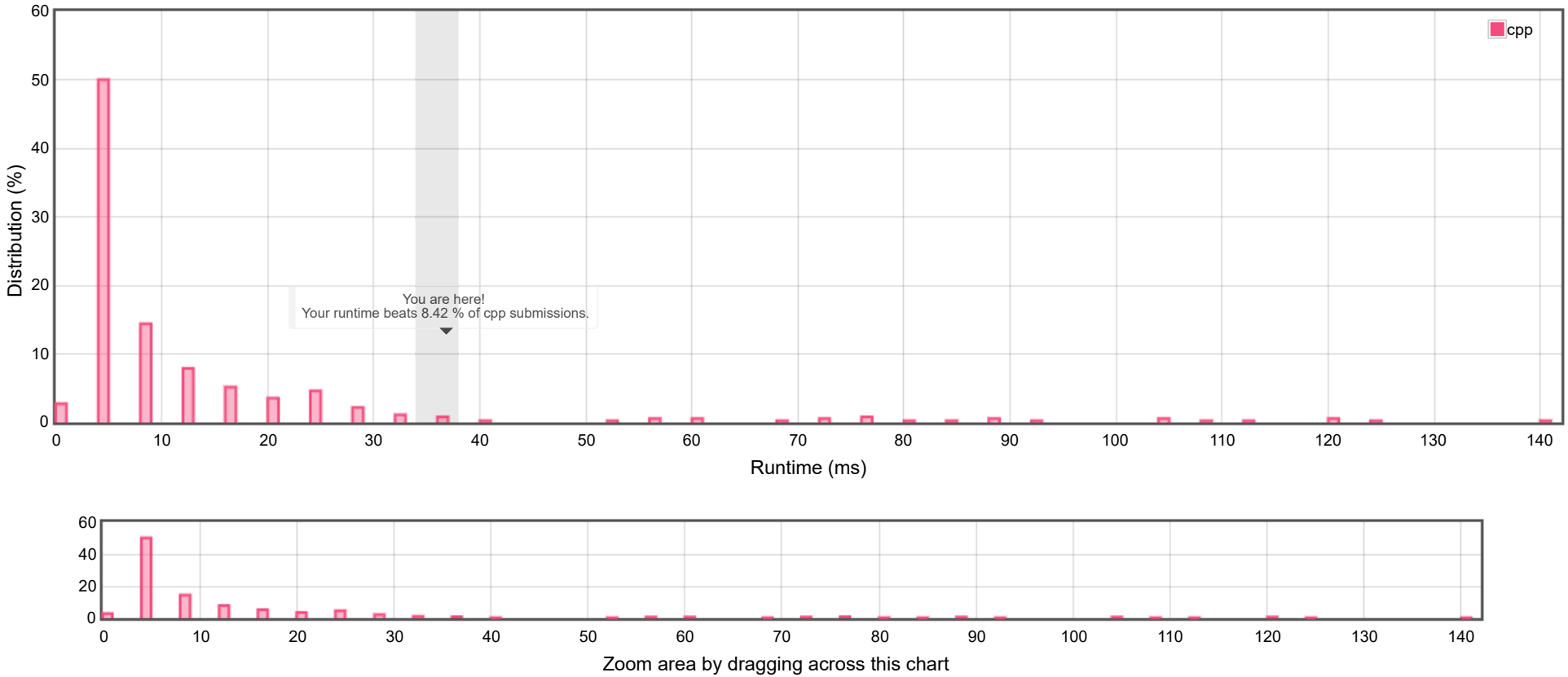
32 / 32 test cases passed.

Status: Accepted

Runtime: **36 ms**

Submitted: **35 minutes ago**

Accepted Solutions Runtime Distribution



Invite friends to challenge **Sliding Puzzle**

Submitted Code: 35 minutes ago

Language: cpp

Edit Code

```cpp
class Solution {
public:
    int slidingPuzzle(vector<vector<int>>& board) {
        int count = 0;
        queue<vector<vector<int>>> q;
        map<vector<vector<int>>, bool> visited;
        q.push(board);
        while(!q.empty()) {
            vector<vector<vector<int>>> boards;
            while (!q.empty()) {
                boards.push_back(q.front());
                q.pop();
            }
            for (int i = 0; i < boards.size(); i++) {
                vector<vector<int>> b(boards[i]);
                if (goal(b)) {
                    return count;
                }
                visited[b] = true;
                int zero_i = 0, zero_j = 0;
                bool f = false;
                for (zero_i = 0; zero_i < 2; zero_i++) {
                    for (zero_j = 0; zero_j < 3; zero_j++) {
                        if (b[zero_i][zero_j] == 0) {
                            f = true;
                            break;
                        }
                    }
                    if (f) {
                        break;
                    }
                }
                if (zero_i == 0) {
                    vector<vector<int>> tmp(b);
                    mSwap(tmp, zero_i, zero_j, zero_i + 1, zero_j);
                    if (!visited[tmp])
                        q.push(tmp);
                } else {
                    vector<vector<int>> tmp(b);
                    mSwap(tmp, zero_i, zero_j, zero_i - 1, zero_j);
                    if (!visited[tmp])
                        q.push(tmp);
```

```
                }
                if (zero_j == 0) {
                    vector<vector<int>> tmp(b);
                    mSwap(tmp, zero_i, zero_j, zero_i, zero_j + 1);
                    if (!visited[tmp])
                        q.push(tmp);
                } else if (zero_j == 1) {
                    vector<vector<int>> tmp(b);
                    mSwap(tmp, zero_i, zero_j, zero_i, zero_j - 1);
                    if (!visited[tmp])
                        q.push(tmp);
                    mSwap(tmp, zero_i, zero_j, zero_i, zero_j - 1);
                    mSwap(tmp, zero_i, zero_j, zero_i, zero_j + 1);
                    if (!visited[tmp])
                        q.push(tmp);
                } else {
                    vector<vector<int>> tmp(b);
                    mSwap(tmp, zero_i, zero_j, zero_i, zero_j - 1);
                    if (!visited[tmp])
                        q.push(tmp);
                }
            }
            count++;
        }
        return -1;
    }
    void mSwap(vector<vector<int>> &arr, int row1, int col1, int row2, int col2) {
        int tmp = arr[row1][col1];
        arr[row1][col1] = arr[row2][col2];
        arr[row2][col2] = tmp;
    }
    bool goal(vector<vector<int>> &board) {
        return board[0][0] == 1 && board[0][1] == 2 && board[0][2] == 3 &&
            board[1][0] == 4 && board[1][1] == 5 && board[1][2] == 0;
    }
    void printarr(vector<vector<int>> &arr) {
        for (int i = 0; i < 2; i++) {
            for (int j = 0; j < 3; j++) {
                printf("%d ", arr[i][j]);
            }
            printf("\n");
        }
    }
};
```

Back to problem (/problems/sliding-puzzle/)