

w Go Run ...



Untitled (Workspace)

t.cpp • merge sorted array • even.cpp

merge sorted array1.cpp 1 •

valid sudoku

c++programme > first.cpp > valid sudoku.cpp > ...

```
1 // valid sudoku problem
2 class solution
3 {
4     public:
5     {
6         bool validbox(vector<vector<char>& board,int sr,int er,int sc,int ec)
7         {
8             //3*3
9             unordered_set<char> st;
10
11             for(int i = sr; i<=er; i++)
12             {
13                 for(int j=sc; j<=ec; j++)
14                 {
15                     if(board[i][j] == '.')
16                         continue;
17                     if(st.find(board[i][j]) != st.end())
18                         return false;
19                     st.insert(board[i][j]);
20                 }
21             }
22         }
23     }
24     bool isvalidsudoku(vector<vector<char>>& board)
25     {
26         // rows 9
27         // rows 9
28
29         // validate rows
```


c++programme > first.cpp > valid sudoku.cpp > ...

```
29 // validate rows
30 for(int row = 0; row < 9; row++)
31 {
32     unordered_set<char> st;
33     for(int col = 0; col < 9; col++)
34     {
35         if(board[row][col] == '.')
36             continue;
37         if(st.find(board[row][col]) != st.end())
38             return false;
39         st.insert(board[row][col]);
40     }
41 }
42
43
44 // validate columns
45
46 for(int col = 0; col < 9; col++)
47 {
48     for(int row = 0; row < 9; row++)
49     {
50         if(board[row][col] == '.')
51             continue;
52         if(st.find(board[row][col]) != st.end())
53             return false;
54         st.insert(board[row][col]);
55     }
56 }
57 // validate the 3*3 boxes
58 for(int sr = 0; sr < 9; sr+=3)
```


c++programme > first.cpp >

valid sudoku.cpp > ...

st.insert(board[row][col]);

}

// validate the 3*3 boxes

for(int sr = 0; sr<9; sr+=3)

{

int er = sr+2;

for(int sc = 0; sc<9; sc+=3)

{

int ec = sc+2;

//sr,er,sc,ec

if(!validbox(board,sr,er,sc,ec))

return false;

}

}

return true;

}



}

merge sorted array even.cpp merge sorted array1.cpp 1 valid sudoku.cpp 2
c++programme > first.cpp > k differ pair in array.cpp > ...

```
1  class solution
2  {
3      public:
4          int findpairs(vector<int>& nums,int k)
5          {
6              if(k < 0)
7                  return 0;
8              unordered_map<int, int> m;
9              for(int n: nums)m[n]++;
10             int cnt = 0;
11             for(auto p : m)
12             {
13                 if((!k && p.second > 1) || (k && m.count(p.first + k))) ++cnt;
14             }
15             return cnt;
16         }
17 };
```