

第七周练习

张睿恒 2024302182001

完成题目：547-简单、684-简单、685-中等、130-中等、765-困难、1627-困难

T1 LeetCode547

```
class Solution {
public:
    int findCircleNum(vector<vector<int>>& isConnected) {
        int n = isConnected.size();
        int ans = 0;
        unordered_set <int> vis;
        for(int i=0;i<n;++i){
            if(vis.find(i) == vis.end()){
                ++ans;
                queue <int> que;
                que.push(i);
                while(!que.empty()){
                    int cur = que.front();
                    que.pop();
                    for(int j=0;j<n;++j){
                        if(vis.find(j) == vis.end() and isConnected[cur][j] == 1){
                            vis.insert(j);
                            que.push(j);
                        }
                    }
                }
            }
        }
        return ans;
    }
};
```

Lee x | Accepted x

← All Submissions

Accepted 114 / 114 testcases passed

ltheng submitted at Nov 19, 2025 18:06

Solution

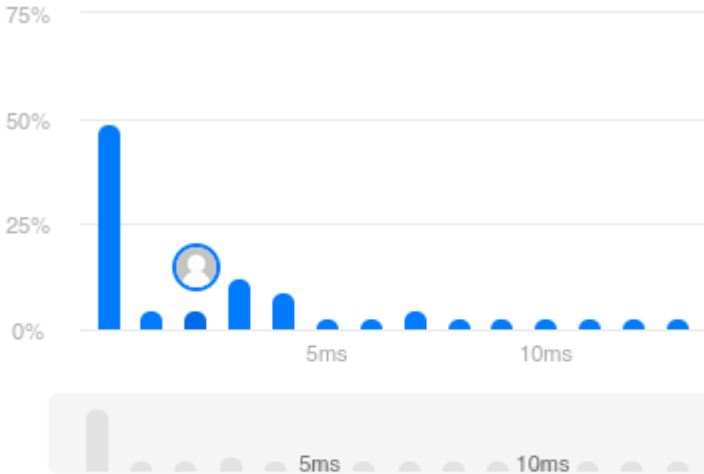
Runtime

2 ms | Beats 46.32%

Analyze Complexity

Memory

20.00 MB | Beats 51.35%



Code | C++

```
class Solution {
public:
    int findCircleNum(vector<vector<int>>& isConnected) {
        int n = isConnected.size();
        int ans = 0;
```

T2 Leetcode 684

```

class Solution {
public:
    vector<int> findRedundantConnection(vector<vector<int>>& edges) {
        unordered_map <int,vector <int> > graph;

        for(const auto& edge : edges){
            int u = edge[0];
            int v = edge[1];

            if(graph.count(u) and graph.count(v) and check(u,v,graph)) return edge;

            graph[u].push_back(v);
            graph[v].push_back(u);
        }

        return {};
    }
private:
    bool check(int u,int v,unordered_map <int,vector <int> >& graph){
        unordered_set <int> vis;
        stack <int> st;
        st.push(u);
        while(!st.empty()){
            int cur = st.top();
            st.pop();
            if(vis.count(cur)) continue;
            vis.insert(cur);
            if(cur == v) return true;
            if(graph.count(cur)){
                for(int neighbour : graph[cur]) st.push(neighbour);
            }
        }
        return false;
    }
};

```

Lee x | Accepted x

← All Submissions [Link](#)

Accepted 39 / 39 testcases passed

📖

[Solution](#)

👤 ltheng

 submitted at Nov 19, 2025 18:24

🕒 Runtime

ⓘ

15 ms | Beats 8.97%

[🔍 Analyze Complexity](#)

💾 Memory

17.92 MB | Beats 8.57%

A bar chart showing the distribution of runtime times for all submissions. The x-axis represents runtime in milliseconds (0 to 20ms), and the y-axis represents the percentage of submissions (0% to 75%). A prominent blue bar at 15ms indicates that a significant portion of submissions achieved this runtime. A circular profile icon is overlaid on the 15ms bar, signifying the user's performance. Below the chart, a horizontal bar shows the relative performance of the user's submission compared to others.

Runtime (ms)	Percentage (%)
15	~58
16	~5
17	~2
18	~1
19	~1
20	~1

Code | C++

```
class Solution {
public:
    vector<int> findRedundantConnection(vector<
        unordered_map <int, vector <int> > graph
```

T3 Leetcode 685

```

class DSU {
    vector<int> size, parent;
public:
    DSU(int n) {
        size.resize(n+1, 1);
        parent.resize(n+1);
        iota(parent.begin(), parent.end(), 0);
    }
    int find(int u) {
        if(parent[u] == u) return u;
        return parent[u] = find(parent[u]);
    }
    void unionBySize(int u, int v) {
        int root_u = find(u);
        int root_v = find(v);
        if(root_u == root_v) return;
        if(size[root_u] < size[root_v]) {
            parent[root_u] = root_v;
            size[root_v] += size[root_u];
        }
        else {
            parent[root_v] = root_u;
            size[root_u] += size[root_v];
        }
    }
};

class Solution {
public:
    vector<int> findRedundantDirectedConnection(vector<vector<int>>& edges) {
        int n = edges.size();
        vector<int> parent(n+1, 0);
        vector<int> candidate1, candidate2;
        for(auto &e : edges) {
            int u = e[0], v = e[1];
            if(parent[v] == 0) parent[v] = u;
            else {
                candidate1 = {parent[v], v};
                candidate2 = {u, v};
                e[1] = 0;
            }
        }
        DSU ds(n+1);
        for(auto &e : edges) {
            int u = e[0], v = e[1];
            if(v == 0) continue;
            if(ds.find(u) == ds.find(v)) {

```

```
        if(candidate1.empty()) return {u, v};  
        return candidate1;  
    }  
    else ds.unionBySize(u, v);  
}  
return candidate2;  
}  
};
```

Lee x | Accepted x

← All Submissions [Link](#)

Accepted 58 / 58 testcases passed

📖

[Solution](#)

👤 ltheng

 submitted at Nov 19, 2025 18:29

🕒 Runtime

ⓘ

2 ms | Beats 34.18%

[🔗 Analyze Complexity](#)

💾 Memory

13.65 MB | Beats 78.82% 🍃

75%
50%
25%
0%

10ms 23ms 39ms 62ms

10ms 23ms 39ms 62ms

Code | C++

```
class DSU {
    vector<int> size, parent;
public:
    DSU(int n) {
        size.resize(n+1, 1);
```

T4 Leetcode130

```

class Solution {
public:
    void markRegion(std::vector<std::vector<char>>& board, int row, int col) {
        int m = board.size(), n = board[0].size();

        if (row < 0 or row >= m or col < 0 or col >= n or board[row][col] != '0')
            return;

        board[row][col] = '*';

        markRegion(board, row + 1, col);
        markRegion(board, row - 1, col);
        markRegion(board, row, col + 1);
        markRegion(board, row, col - 1);
    }

    void solve(vector<vector<char>>& board) {
        int m = board.size(), n = board[0].size();
        for (int row = 0; row < m; row++) {
            if (board[row][0] == '0') markRegion(board, row, 0);
            if (board[row][n - 1] == '0') markRegion(board, row, n - 1);
        }
        for (int col = 1; col < n - 1; col++) {
            if (board[0][col] == '0') markRegion(board, 0, col);
            if (board[m - 1][col] == '0') markRegion(board, m - 1, col);
        }

        for (int row = 0; row < m; row++) {
            for (int col = 0; col < n; col++) {
                char& c = board[row][col];
                if (c == '*') c = '0';
                else if (c == '0') c = 'X';
            }
        }
    }
};

```


Lee x | Accepted x

← All Submissions

Accepted 58 / 58 testcases passed

Solution

ltheng

 submitted at Nov 19, 2025 18:48

⌚ Runtime

0 ms | Beats 100.00% 🏆

🔍 Analyze Complexity

💾 Memory


13.93 MB | Beats 80.97% 🏆

75%

50%

25%

0%



5ms

10ms

Code | C++

```
class Solution {
public:
    void markRegion(std::vector<std::vector<ch
        int m = board.size(), n = board[0].size
```

T5 Leetcode765

```

class DSU{
public:
vector<int>size,parent;
DSU(int n){
    size.resize(n+1,1);
    parent.resize(n+1);
    for(int i =0;i<n+1;i++) parent[i]=i;
}
int findultiparent(int u){
    if(parent[u] == u) return u;
    return parent[u] = findultiparent(parent[u]);
}
void unionBysize(int u,int v){
    int ulp_u = findultiparent(u);
    int ulp_v = findultiparent(v);

    if(ulp_u == ulp_v) return;

    if(size[ulp_u]>size[ulp_v]){
        size[ulp_u]+=size[ulp_v];
        parent[ulp_v] = ulp_u;
    }
    else{
        size[ulp_v]+=size[ulp_u];
        parent[ulp_u] = ulp_v;
    }
}
};

class Solution {
public:
    int minSwapsCouples(vector<int>& row) {
        int n = row.size();
        unordered_map<int,int>mpp;
        for(int i =0;i<n;i++) mpp[row[i]] = i/2;

        DSU ds(n/2);
        int swaps = 0;
        for(int i =0;i<n;i++){
            if(row[i]%2==0){
                int u = row[i],v = row[i]+1;
                if(ds.findultiparent(mpp[u])!=ds.findultiparent(mpp[v])){
                    swaps++;
                    ds.unionBysize(mpp[u],mpp[v]);
                }
            }
            else{
                int u = row[i],v = row[i]-1;

```

```
        if(ds.findultiparent(mpp[u])!=ds.findultiparent(mpp[v])){
            swaps++;
            ds.unionBysize(mpp[u],mpp[v]);
        }
    }
    return swaps;
}

};
```

Lee x | Accepted x

← All Submissions [Link](#)

Accepted 56 / 56 testcases passed

ltheng submitted at Nov 19, 2025 19:02

📖

Solution

⌚ Runtime ⓘ

1 ms | Beats 17.23%

🔗 Analyze Complexity

💻 Memory

9.88 MB | Beats 80.54% 🏆

Runtime Range	Percentage
0-1ms	~85%
1-2ms	~10%
2-3ms	~2%
3-4ms	~2%
4-5ms	~1%

Code | C++

```
class DSU{
public:
    vector<int>size,parent;
    DSU(int n){
        size.resize(n+1,1);
```

T6 Leetcode1627

```

class DSU{
    vector<int> rank, parent;
    int n;
    public:
    DSU(int n){
        this->n=n;
        rank.resize(n+1,0);
        parent.resize(n+1,0);
        for(int i=0;i<n;i++){
            parent[i]=i;
        }
    }

    int ultpar(int node){
        if(parent[node]==node)return node;
        else return parent[node]=ultpar(parent[node]);
    }

    void unionbyrank(int u, int v){
        int ult_u=ultpar(u);
        int ult_v=ultpar(v);
        if(ult_u==ult_v)return ;

        if(rank[ult_u]<rank[ult_v]) parent[ult_u]=ult_v;
        else if(rank[ult_v]<rank[ult_u]) parent[ult_v]=ult_u;

        else {
            parent[ult_v]=ult_u;
            rank[ult_u]++;
        }
    }
};

class Solution {
public:
    vector<bool> areConnected(int n, int threshold, vector<vector<int>>& queries) {
        int q=queries.size();
        auto dsu=DSU(n);

        for(int i=threshold+1; i<=n; i++){
            int num=i, mult=2;
            while(num<=n){
                dsu.unionbyrank(num,i);
                num=i*mult;
                mult++;
            }
        }
    }
};

```

```
    }

    vector<bool> ans;
    for(int i=0;i<queries.size();i++){
        int u=queries[i][0], v=queries[i][1];
        if(dsu.ultpar(u)==dsu.ultpar(v)) ans.push_back(true);

        else ans.push_back(false);
    }
    return ans;
}

};
```

Accepted 67 / 67 testcases passed

 **ltheng** submitted at Nov 19, 2025 19:10

 **Solution**

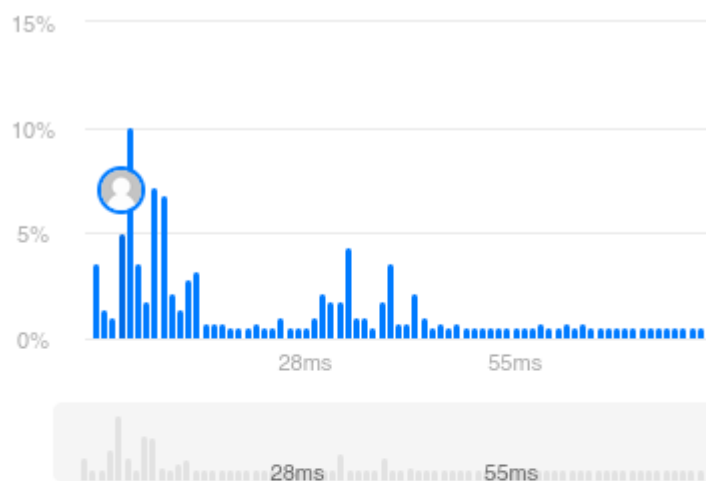
🕒 Runtime ⓘ

3 ms | Beats **93.86%** 🏆

 [Analyze Complexity](#)

💻 Memory

70.27 MB | Beats **63.54%** 🏆



Code | C++

```
//DSU CODE
class DSU{
    vector<int> rank, parent;
    int n;
```