



LIVE

What others are solving.

LIST

List all your submissions.

TRIED

Problems not solved yet.

FAQS

Need help using the tool?

ANSWERS

What does this mean?

SOURCE CODE

[EDIT & SUBMIT](#)

VISUALIZE THE SOURCE CODE OF YOUR SUBMISSION, PLUS SOME EXTRA DETAILS.

SUBMISSION # 25466896

PROBLEM: 1082 - Connected Components
ANSWER: **Accepted**
LANGUAGE: C++17 (g++ 7.3.0, -std=c++17 -O2 -lm) [+0s]
RUNTIME: 0.314s
FILE SIZE: 2.89 KB
MEMORY: -
SUBMISSION: 11/29/21, 3:51:49 AM

SOURCE CODE

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  class Graph{
5      public:
6          map<int, short> visited_vertexes;
7          map<int, vector<std::map<int, int>>> adjacent_vertexes;
8          int total_weight = 0;
9          void add_edge(int vertex1, int vertex2, int weight);
10         void clear_visited_vertexes();
11     };
12
13
14
15 void Graph::add_edge(int vertex1, int vertex2, int weight){
16     std::map<int, int> temp;
17     temp = {{vertex2, weight}};
18     this->adjacent_vertexes[vertex1].push_back(temp);
19     temp = {{vertex1, weight}};
20     this->adjacent_vertexes[vertex2].push_back(temp);
```

FORUM

Get help to solve problems.

```
23
24
25 void Graph::clear_visited_vertexes(){
26     this->visited_vertexes.clear();
27 }
28
29
30 void dfs_tag_vertexes(Graph *g, int current_vertex, int parent_vertex);
31
32 void _dfs_tag_vertexes(Graph *g, int current_vertex, int parent_vertex, std::vector<int> *arr);
33
34
35 int main(){
36     int N, V, E, e1, e2;
37     char aux;
38
39     cin >> N;
40
41     for(int i = 0; i < N; i++){
42         short *array_aux = (short*)calloc(N, sizeof(short));
43         Graph *g = new Graph();
44
45         int componentes = 1;
46
47         std::cout << "Case #" << i+1 << ":\n";
48         std::cin >> V >> E;
49
50         for(int j = 0; j < E; j++){
51             cin >> aux;
52             e1 = aux - 'a';
53             array_aux[e1] = 1;
54
55             std::cin >> aux;
56             e2 = aux - 'a';
57             array_aux[e2] = 1;
58
59             g->add_edge(e1, e2, 0);
60         }
61         for(int i = 0; i < V; i++)
62             if(array_aux[i] == 0)
63                 g->add_edge(i, i, 0);
64
65
66         dfs_tag_vertexes(g, g->adjacent_vertexes.begin()->first, -1);
67         std::cout << "\n";
68
69         for(auto k : g->adjacent_vertexes){
70             if(g->visited_vertexes[k.first] == 0){
```

```

72         dfs_tag_vertexes(g, k.first, -1);
73         cout << "\n";
74     }
75 }
76
77 cout << componentes << "\n connected components\n\n";

```



Hi, RODRIGO JESUS SANTISTEBAN PACHARI
rsantisteban@unsa.edu.pe

[HOME](#)
[PROFILE](#)
[NEWS](#)

352

[INSTRUCTORS](#)
[ACADEMIC](#)
[CONTESTS](#)
[FORUM](#)
[PROBLEMS](#)
[SUBMISSIONS](#)
[RANKS](#)
[SIGN OUT](#)

```

81     }
82
83     return 0;
84 }
85
86 void dfs_tag_vertexes(Graph *g, int current_vertex, int parent_vertex){
87
88     std::vector<int> array_aux;
89
90     _dfs_tag_vertexes(g, current_vertex, parent_vertex, &array_aux);
91
92     std::sort(array_aux.begin(), array_aux.end());
93
94     for(auto i : array_aux)
95         std::cout << (char)(i + 'a') << ", ";
96
97 }
98
99
100 void _dfs_tag_vertexes(Graph *g, int current_vertex, int parent_vertex, std::vector<int> *arr){
101     if(g->visited_vertexes[current_vertex] == 2)
102         return;
103
104     g->visited_vertexes[current_vertex] = 1;
105
106     arr->push_back(current_vertex);
107
108
109     for(auto i : g->adjacent_vertexes[current_vertex])
110         for(auto j : i)
111             if(j.first != parent_vertex && g->visited_vertexes[j.first] != 1)
112                 _dfs_tag_vertexes(g, j.first, current_vertex, arr);
113
114     g->visited_vertexes[current_vertex] = 2;
115
116     return;
117 }

```

