CSES Problem Set

Planets and Kingdoms

Submission details

Task:	Planets and Kingdoms
Sender:	Rodry
Submission time:	2021-11-28 01:46:50
Language:	C++11
Status:	READY
Result:	ACCEPTED

Test results

test	verdict	time	
#1	ACCEPTED	0.01 s	»
#2	ACCEPTED	0.01 s	»
#3	ACCEPTED	0.01 s	»
#4	ACCEPTED	0.01 s	»
#5	ACCEPTED	0.01 s	»
#6	ACCEPTED	0.21 s	»
#7	ACCEPTED	0.21 s	»
#8	ACCEPTED	0.21 s	»
#9	ACCEPTED	0.21 s	»
#10	ACCEPTED	0.21 s	»

Code

```
//https://cses.fi/problemset/task/1683/

#include <bits/stdc++.h>
```


Your submissions

2021-11-28 01:46:50 ✓

2021-11-28 01:35:59 ✗

```
#define INF 99999
#define MAX 999

using namespace std;

void busqueda_profundidad(vector<vector<int>>& matriz_adyacencia, vector<bool>& v
    if (visitados[act])
        return;

    visitados[act] = true;

    for (int a : matriz_adyacencia[act])
        busqueda_profundidad(matriz_adyacencia, visitados, orden, a);

    orden.push_back(act);
}

void busqueda_profundidad2(const vector<vector<int>>& matriz_adyacencia, vector<i
    if (componentes[act])
        return;

    componentes[act] = k;

    for (int a : matriz_adyacencia[act]) {
        busqueda_profundidad2(matriz_adyacencia, componentes, a, k);
    }
}

int main(){

    int n, m;
    int a, b;
    cin>>n; cin>>m;

    vector<vector<int>> matriz_adyacencia(n + 1);
        vector<vector<int>> reversa(n + 1);

    //Inputs
    for (int i = 0; i < m; i++) {
        cin>>a; cin>>b;
        matriz_adyacencia[a].push_back(b);
        reversa[b].push_back(a);
    }

        //Vectores auxiliares
```

```
    vector<int> orden;
    vector<bool> visitados(n + 1, false);


    for (int i = 1; i <= n; i++) {
        if (visitados[i])
            continue;
        busqueda_profundidad(matriz_adyacencia, visitados, orden, i);
    }

    vector<int> componentes(n + 1);
    reverse(orden.begin(), orden.end());

    int k = 0;
    for (auto a : orden) {
        if (componentes[a])
            continue;
        k++;
        busqueda_profundidad2(reversa, componentes, a, k);
    }

        //Salida
    cout<<k<<endl;
    for (int i = 1; i <= n; i++)
        cout<<componentes[i]<<" ";
    cout<<endl;
}
```

Share code to others

Test details

Test 1

Verdict: ACCEPTED

input

```
10 20
4 5
10 7
6 1
6 5
...
```

correct output

```
4
4 3 1 1 2 1 1 1 1 1
```

user output

```
4
4 3 1 1 2 1 1 1 1 1
```

Test 2

Verdict: ACCEPTED

input

```
10 20
5 6
1 2
7 9
7 3
...
```

correct output

```
2
1 1 1 2 1 1 1 1 1 1
```

user output

```
2
1 1 1 2 1 1 1 1 1 1
```

Test 3

Verdict: ACCEPTED

input

```
10 20
1 6
```

```
10 1
1 10
9 10
...
```

correct output

```
3
3 2 1 3 3 3 3 3 3 3
```

user output

```
3
3 2 1 3 3 3 3 3 3 3
```

Test 4

Verdict: ACCEPTED

input

```
10 20
4 7
10 6
10 5
10 7
...
```

correct output

```
7
2 4 3 1 5 7 5 6 5 5
```

user output

```
7
2 4 3 1 5 7 5 6 5 5
```

Test 5

Verdict: ACCEPTED

input

```
10 20
8 1
7 6
6 4
3 1
...
```

correct output

```
3
3 3 3 3 3 3 3 3 1 2
```

user output

```
3
3 3 3 3 3 3 3 3 1 2
```

Test 6

Verdict: ACCEPTED

input

```
100000 200000
32402 49159
41650 12290
90019 96038
12320 82053
...
```

correct output

```
43755
26333 26333 40691 34351 26331 ...
```

user output

```
43755
26333 26333 40691 34351 26331 ...
```

Test 7

Verdict: ACCEPTED

input

```
100000 200000
98891 58773
74281 97370
25400 8211
25600 1357
...
```

correct output

```
43634
26308 26307 26306 43439 26305 ...
```

user output

```
43634
26308 26307 26306 43439 26305 ...
```

Test 8

Verdict: ACCEPTED

input

```
100000 200000
30442 46106
83330 48942
25229 50273
33345 72005
...
```

correct output

```
43498
43496 26032 26031 32415 26033 ...
```

user output

```
43498
43496 26032 26031 32415 26033 ...
```

Test 9

Verdict: ACCEPTED

input

```
100000 200000
45422 77478
6800 88602
9724 59882
20954 36466
...
```

correct output

```
44087
26407 26407 26407 26404 26407 ...
```

user output

```
44087
26407 26407 26407 26404 26407 ...
```

Test 10

Verdict: ACCEPTED

input

```
100000 200000
77767 73183
30807 58373
23969 94613
27280 75565
...
```

correct output

```
43903
```

```
43903 43902 26255 26256 29892 ...
```

user output

```
43903
43903 43902 26255 26256 29892 ...
```