

Раздел «Алгоритмы» . ArticulationPointsCPP :

Поиск точек раздела: реализация на C++

- Поиск точек раздела: реализация на C++
 - Код
 - Ссылки

Код

```
/*
 * Поиск точек раздела в неориентированном графе.
 * Даниил Швед, 2008. МФТИ.
 * mailto: danshved [no-spam] gmail.com
 */
#include <stdio.h>
#include <algorithm>
#include <vector>
using namespace std;

typedef vector<int> VInt;
typedef vector<VInt> VVInt;
typedef VInt::iterator VIter;
typedef vector<bool> VBool;

VVInt graph;
VInt colors, enter, leave, low, parents;
int myTime = 0;
VBool articulation;

/*
 * Поиск в глубину, вычисление функций enter. leave и low,
 * и здесь же определение точек раздела
 */
void visit(int u) {
    int childrenCount = 0;
    colors[u] = 1;
    low[u] = enter[u] = ++myTime;

    for(VIter it = graph[u].begin(); it != graph[u].end(); it++)
        if(colors[*it] == 0) {
            parents[*it] = u;
            visit(*it);
            low[u] = min(low[u], low[*it]);

            if(parents[u] != -1 && low[*it] >= enter[u])
                articulation[u] = true;
            childrenCount++;
        } else if(colors[*it] == 1 && *it != parents[u]) {
            low[u] = min(low[u], enter[*it]);
        }

    if(parents[u] == -1)
        articulation[u] = (childrenCount >= 2);

    colors[u] = 2;
    leave[u] = ++myTime;
}

/*
 * Во входном потоке: числа n и m, затем m описаний ребер.
```

Поиск

Раздел «Алгоритмы»

[Главная](#)
[Форум](#)
[Ссылки](#)
[EI Judge](#)

Инструменты:

[Поиск](#)
[Изменения](#)
[Index](#)
[Статистика](#)

Разделы

[Информация](#)
[Алгоритмы](#)
[Язык Си](#)
[Язык Ruby](#)
[Язык](#)
[Ассемблера](#)
[EI Judge](#)
[Парадигмы](#)
[Образование](#)
[Сети](#)
[Objective C](#)

[Login>>](#)

```
* В выходном потоке: номера точки раздела
*/
int main() {
    int n, m, i;

    // Считаем граф
    scanf("%d%d", &n, &m);
    graph.resize(n);
    while(m--) {
        int from, to;
        scanf("%d%d", &from, &to);
        graph[from - 1].push_back(to - 1);
        graph[to - 1].push_back(from - 1);
    }

    // Запустим поиск в глубину
    colors.assign(n, 0);
    enter.resize(n);
    leave.resize(n);
    low.resize(n);
    parents.assign(n, -1);
    articulation.assign(n, false);
    for(i = 0; i < n; i++)
        if(colors[i] == 0)
            visit(i);

    // Выведем список точек раздела
    for(i = 0; i < n; i++)
        if(articulation[i])
            printf("%d ", i + 1);
    printf("\n");

    return 0;
}
```

Ссылки

- Поиск мостов, точек раздела и двусвязных компонент (теория)

-- DanielShved - 27 Mar 2008

Copyright © 2003-2022 by the contributing authors.