

Санкт-Петербургский национальный исследовательский университет ИТМО

Факультет систем управления и робототехники

Алгоритмы и структуры данных

Отчёт по лабораторной работе №18 (1160. Network)

Преподаватель: Тропченко А. А.

Выполнил: Марухленко Д. С.

Группа: R3235

Санкт Петербург, 2021г.

1. Цель работы

Решить задачу №1160 на платформе Timus Online Judge

<https://acm.timus.ru/problem.aspx?space=1&num=1160>

2. Задача

Условие

Andrew is working as system administrator and is planning to establish a new network in his company. There will be N hubs in the company, they can be connected to each other using cables. Since each worker of the company must have access to the whole network, each hub must be accessible by cables from any other hub (with possibly some intermediate hubs).

Since cables of different types are available and shorter ones are cheaper, it is necessary to make such a plan of hub connection, that the maximum length of a single cable is minimal. There is another problem - not each hub can be connected to any other one because of compatibility problems and building geometry limitations. Of course, Andrew will provide you all necessary information about possible hub connections.

You are to help Andrew to find the way to connect hubs so that all above conditions are satisfied..

Ограничение времени: 2.0 секунды

Ограничение памяти: 64 МБ

Исходные данные

The first line contains two integer: N - the number of hubs in the network ($2 \leq N \leq 1000$) and M — the number of possible hub connections ($1 \leq M \leq 15000$). All hubs are numbered from 1 to N . The following M lines contain information about possible connections - the numbers of two hubs, which can be connected and the cable length required to connect them. Length is a positive integer number that does not exceed 106. There will be no more than one way to connect two hubs. A hub cannot be connected to itself. There will always be at least one way to connect all hubs.

Результат

Output first the maximum length of a single cable in your hub connection plan (the value you should minimize). Then output your plan: first output P - the number of cables used, then output P pairs of integer numbers - numbers of hubs connected by the corresponding cable. Separate numbers by spaces and/or line breaks.

Пример

Исходные данные	Результат
4 6	1
1 2 1	4
1 3 1	1 2
1 4 2	1 3
2 3 1	2 3
3 4 1	3 4
2 4 1	

3. Материалы работы

3.1. Объяснение алгоритма

Задача сводится к использованию алгоритма Краскала на графах. Хабы представлены как вершины графа, а провода – рёбра. При считывании все рёбра записываются в вектор и сортируются. После – рёбра добавляются в граф с присвоением каждой вершине номера, соответствующего номеру подграфа, не соединённого с другими подграфами. В случае объединения двух подграфов им присваивается наименьший из их номеров. Это продолжается до тех пор, пока все вершины графа не будут соединены.

3.2. Код программы.

```
1. #include <iostream>
2. #include <vector>
3. #include <algorithm>
4.
5. using namespace std;
6.
7. struct cable {
8.     unsigned short a, b;
9.     unsigned l;
10. };
11.
12. bool compare(cable a, cable b) {
13.     return a.l < b.l;
14. }
15.
16. int main() {
17.
18.     unsigned short n, m;
19.
20.     cin >> n >> m;
21.     vector<cable> v;
22.
23.     unsigned short a, b;
24.     unsigned l;
25.     for (int i = 0; i < m; ++i) {
26.         cin >> a >> b >> l;
27.         v.push_back((cable){a-1, b-1, l});
28.     }
29.
30.     sort(v.begin(), v.end(), compare);
31.
32.     short used_points[n];
33.     short used_vectors[m];
34.
35.     fill_n(used_points, n, -1);
36.     fill_n(used_vectors, m, 0);
37.
38.     unsigned group_number = 0;
39.     unsigned max = 0;
40.     unsigned counter = 0;
41.     for (unsigned short i = 0; i < m; i++){
42.         if (used_points[v[i].a] != -1 && used_points[v[i].a] ==
used_points[v[i].b]) continue;
43.         else if (used_points[v[i].a] == -1 && used_points[v[i].b] == -1 ){
44.             ++group_number;
45.             used_points[v[i].a] = group_number;
46.             used_points[v[i].b] = group_number;
47.         }else if ((used_points[v[i].a] != -1 && used_points[v[i].b] == -1
)|| (used_points[v[i].a] == -1 && used_points[v[i].b] != -1 )){
48.             int groupnumber = used_points[v[i].a]+used_points[v[i].b]+1;
49.             used_points[v[i].a] = groupnumber;
50.             used_vectors[v[i].b] = groupnumber;
51.         }else{
52.             int mingroupnumber;
53.             int maxgroupnumber;
54.             if (used_points[v[i].b]>used_points[v[i].a]) {
```

```

55.             mingroupnubmer = used_points[v[i].a];
56.             maxgroupnubmer = used_points[v[i].b];
57.         } else {
58.             mingroupnubmer = used_points[v[i].b];
59.             maxgroupnubmer = used_points[v[i].a];
60.         }
61.         for (unsigned short j = 0; j < n; j++)
62.             if(used_points[j] == maxgroupnubmer) used_points[j] =
mingroupnubmer;
63.         }
64.         used_vectors[i] = 1;
65.         max = v[i].l;
66.         ++counter;
67.     }
68.
69.     cout << max << endl << counter << endl;
70.
71.     for (unsigned short j = 0; j < m; j++)
72.         if (used_vectors[j])
73.             cout << v[j].a+1<< " " << v[j].b+1 << endl;
74.     return 0;
75. }

```

4. Результат выполнения и ссылка на репозиторий GitHub

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
9302158	19:56:39 4 апр 2021	Daniil Marukhlenko	1160	G++ 9.2 x64	Accepted		0.046	700 КБ

https://github.com/japersik/algorithms_and_data_structures/



5. Вывод

Работа выполнена, задача решена с использованием алгоритма Краскала на графах.