added: 10 Jun 2008 22:05
Edited: 3 May 2010 23:29

A minimum spanning tree. Kruskal's algorithm with a system of disjoint sets

Formulation of the problem and a description of Kruskal's algorithm see. here .

There will consider the implementation using the data structure "system of disjoint sets" (DSU) , which will reach the asymptotic behavior of **O (Log M N)** .

Description

Just as in the simple version of Kruskal's algorithm, we can sort all edges by non-decreasing weight. Then put each node in your tree (ie their set) by calling the DSU MakeSet - it will take in the amount of O (N). Loop through all edges (in the sort order), and for each edge in O (1) to determine, whether it belongs to the ends of the different trees (using two calls FindSet O (1)). Finally, the union of two trees will be calling the Union - also in O (1). Overall, we obtain the asymptotic behavior of O (M log N + N + M) = O (M log N).

Implementation

To reduce the volume of code and carry out all operations are not as separate functions, and directly in the code of Kruskal's algorithm.

Here we will use a randomized version of the DSU.

```
vector <int> p (n);

int dsu_get (int v) {
        return (v == p [v])? v: (p [v] = dsu_get (p [v]));
}

void dsu_unite (int a, int b) {
        a = dsu_get (a);
        b = dsu_get (b);
        if (rand () & 1)
                swap (a, b);
        if (a! = b)
                p [a] = b;
}

... Function main (): ...

int m;
vector <pair <int, pair <int, int>>> g; // Weight - the top 1 - top 2
Graph reading ... ...

int cost = 0;
vector <pair <int, int>> res;

sort (g.begin (), g.end ());
p.resize (n);
for (int i = 0; i <n; ++ i)
        p [i] = i;
for (int i = 0; i <m; ++ i) {
        int a = g [i] .second.first, b = g [i] .second.second, l = g [i] .first;
        if (dsu_get (a)! = dsu_get (b)) {
                cost + = l;
                res.push_back (g [i] .second);
                dsu_unite (a, b);
        }
}
```

Sergey Blinov • 2 года назад

Я думаю, что стоит указать, откуда получилась сложность O(M log N). Если смотреть по изложению, то получается O(M log M + N + M) = O(M log M).
Но т.к. M < N * N, то асимптотика получается O(M log N^2) =
O(2M log N) = O(M log N).
Если это где-то объяснено, то поставьте пожалуйста ссылку, потому что неопытный читатель может не догадаться.