最佳與次佳網路建構成本

時間限制：2秒

Solution for second MST:

The problem asks for the difference of the costs between an MST and a second MST. The key property to find the second MST is that it differs from MST by exactly one edge. In this problem, all edge weights are distinct. Let T be an MST. For each non-tree edge (u,v), we need to find the largest edge e of the unique uv-path on T. The final solution is min{w(u,v)-w(e)}, where the minimum is taken for all nontree edges. For a vertex s, by a simple tree traversal we can find the largest edge weight from s to every vertex v. Therefore it gives us an O(n^2)-time algorithm, where n is the number of vertices.

A better algorithm arranges all the nontree-edge queries as follows.

1. Root the MST T at an arbitrary vertex.
2. Find the lowest common ancestor (lca) of u and v for each nontree edge (u,v).
3. Perform a post-order traversal and each query is performed when visiting its lca.
4. When query (u,v) at their lca r: we find the largest edges from u and v to r by the following procedure, where a is an ancestor or u.

int bfind(int u,int a){

if (u==a) return 0;

if (set[u]!=a) {

big[u]=MAX2(big[u],bfind(set[u],a));

set[u]=a;

}

return big[u];

}

The time complexity is the same as find-and-union. All the lca can be found by using **Tarjan's off-line lowest common ancestors algorithm** (see Wiki), also in the same time complexity. So the total time complexity is bounded by the step of finding MST.

Test data:

For subtask 1 and 2: they are special inputs one can solve the problem without knowing MST. Provided programs also check the validity of inputs.

For other subtasks: two kinds of inputs are given: random edges and k-path MST (the MST consists of k paths joined to a same vertex, k=2,3,4). The k-path-MST data has an MST of very large diameter and the edge weights are specially designed such that the second MST uses a very large edge. A program will get WA if it only queries some small nontree edges; and get TLE if it does not use the lifting technique as in find-and-union.