

Sublinear Algorithms: Minimum Spanning Forest with fixed Parameters

Usually a graph algorithm requires $\Omega(|E|)$ time, because the algorithm has to read the entire input, i.e. every edge of the graph. As a result classical graph algorithms can often not be used for extremely large graphs. Interestingly, for many problems we do not need to read the entire graph in order to give an approximate solution.

The task

Write a program that estimates the weight of a minimum spanning forest, such that:

- The program only needs to know small fraction of the input.
- The program outputs a 1.2-approximation of the weight of a minimum spanning forest. This means $\text{opt}/1.2 \leq \text{output} \leq 1.2\text{opt}$.
- Even though the optimal value is an integer, your program is allowed to return a float value as approximation. E.g. you could output 9.5 if you don't know if the optimal value is 9 or 10.
- The program is allowed to be randomized.

You may assume the following about the input graphs:

- The graph is undirected.
- The input graphs will have less than 2^{31} nodes. So you do not have to worry about integer overflows when using signed 32 bit integers.
- The input graphs will have small degree, i.e. bounded by some constant.
- The weights C_{ij} of the edges are natural numbers and are at most 4, so $1 \leq C_{i,j} \leq 4$ for every edge (i,j) .
- You may assume the optimal spanning forest has weight at least n .
- The nodes are indexed as $0, 1, \dots, n-1$.

Interaction

As you are supposed to only read a small fraction of the input, your program is not given the entire graph at once. At first, you are only given the number of nodes n . Your program is then allowed to ask Kattis for more data.

Asking for data:

If your program wants to know the neighbors of node number V , then it only has to print the number V to stdout. Kattis will then give the neighbor information on stdin. The format of the answer is the following: The answer is given by a single line, starting with the number of neighbors. The rest of the line is a list of neighbors and the cost of the respective edges. For instance `3 5 2 9 4 1 1` means the node V has 3 neighbors 5, 9 and 1. The cost of these edges is $c_{v,5}=2$, $c_{9,5}=4$ and $c_{v,1}=1$.

Reporting your result

If your program wants to report a result, it should print `end` followed by the weight of the spanning forest. For instance `end 10.312`, when your program computed a weight of 10.312.

Communication example

Your program's output (stdout)	Kattis' input/answer (stdin)	interpretation
	10	The graph has 10 nodes.
7		Your program asks for the neighbors of node 7.
	3 1 3 6 2 3 1	Node 7 has 3 incident edges: (7,1) with weight 3, (7,6) with weight 2 and (7,3) with weight 1.
1		Your program asks for the neighbors of node 1.
	1 7 3	Node 1 has only one neighbor: 7 and the corresponding edge has weight 3.

2

0

end 4.12

Your program asks for the neighbors of node 2.

Node 2 has no neighbors

Your program thinks 4.12 is a 1.2-approximation for the weight of the minimum spanning forest. Your program should exit now.

Remarks on performance:

The 'model solution' was written in Python 2.7. You do not need to write your program in C++ or other low-level languages in order to pass the runtime limit. Instead of runtime complexity you should focus on getting a good approximation while reading only a small part of the graph.