



Central Europe Regional Contest 2018

Trees Gump

`gump.c`, `gump.cpp`, `Gump.java`, `gump.py`

The huge trees in the Jumbarian jungle are strategically very important. The Jumbarian army headquarters have selected N trees which are rather close to each other, and decided that a tree house will be built on each of these trees and it will be occupied by a single army unit. Movement of people and material between the tree tops will be supported by the system of bi-directional zip lines connecting pairs of tree tops. For safety reasons, no two zip lines can cross each other (when observed from a satellite).

N units have already been chosen and a list of pairs of these units has been created. Units in a pair are expected to operate on the opposite ends of one zip line. The number of pairs in the list is one less than the number of units. It turns out the list ensures the connectedness of the area, i.e., after the units will be deployed to the trees, each unit will be able to reach any other unit using only zip lines that appear on the list.

All that remains to put this scheme to work is to install the zip lines between tree tops in such a way that will allow to deploy the units in accordance with the above rules.

Input Specification

The first line contains a number N of tree tops and army units ($1 \leq N \leq 1000$). Both the tree tops and the army units are labeled $0, 1, \dots, N-1$. The next $N-1$ lines contain the list of pairs of units. Each line contains labels of two units expected to operate on the opposite ends of a single zip line. The next N lines give the coordinates of the selected tree tops. The $(i+1)$ -th of these lines contains two numbers X_i and Y_i ($0 \leq X_i, Y_i \leq 10^9$), the coordinates of the tree top labeled i . No three tree tops lie on a single line.

Output Specification

Output a list of all pairs of trees which are to be connected by a zip line. The list consists of $N-1$ lines, each line contains two labels of different trees.

If there are multiple solutions to the problem, print any of them.

Sample Input 1

5
0 1
1 2
2 3
3 4
0 0
9 9
2 3
3 2
7 8

Output for Sample Input 1

0 3
3 1
1 4
4 2

Sample Input 2

3
1 2
0 2
0 0
1 1
2 3

Output for Sample Input 2

0 1
1 2