

USA Computing Olympiad



OVERVIEW

DETAILS / FAQ

TRAINING

HISTORY

RESOURCES

USACO 2025 JANUARY CONTEST, PLATINUM PROBLEM 1. DFS ORDER

[Return to Problem List](#)

Contest has ended.

[Log in to allow submissions in analysis mode](#)

English (en) ▾

Bessie has a simple undirected graph with vertices labeled $1 \dots N$ ($2 \leq N \leq 750$). She generates a depth-first search (DFS) order of the graph by calling the function `dfs(1)`, defined by the following C++ code. Each adjacency list (`adj[i]` for all $1 \leq i \leq N$) may be permuted arbitrarily before starting the depth first search, so a graph can have multiple possible DFS orders.

```
vector<bool> vis(N + 1);
vector<vector<int>> adj(N + 1); // adjacency list
vector<int> dfs_order;

void dfs(int x) {
    if (vis[x]) return;
    vis[x] = true;
    dfs_order.push_back(x);
    for (int y : adj[x]) dfs(y);
}
```

You are given the initial state of the graph as well as the cost to change the state of each edge. Specifically, for every pair of vertices (i, j) satisfying $1 \leq i < j \leq N$, you are given an integer $a_{i,j}$ ($0 < |a_{i,j}| \leq 1000$) such that

- If $a_{i,j} > 0$, edge (i, j) is not currently in the graph, and can be added for cost $a_{i,j}$.
- If $a_{i,j} < 0$, edge (i, j) is currently in the graph, and can be removed for cost $-a_{i,j}$.

Determine the minimum total cost to change the graph so that $[1, 2, \dots, N]$ is a possible DFS ordering.

INPUT FORMAT (input arrives from the terminal / stdin):

The first line contains N .

Then $N - 1$ lines follow. The $j - 1$ th line contains $a_{1,j}, a_{2,j}, \dots, a_{j-1,j}$ separated by spaces.

OUTPUT FORMAT (print output to the terminal / stdout):

The minimum cost to change the graph so that $[1, 2, \dots, N]$ is a possible DFS ordering.

SAMPLE INPUT:

```
4
1
2 3
40 6 11
```

SAMPLE OUTPUT:

```
10
```

Initially, the graph contains no edges. $(1, 2), (2, 3), (2, 4)$ can be added for a total cost of $1 + 3 + 6$. The graph now has two possible DFS orderings: $[1, 2, 3, 4], [1, 2, 4, 3]$.

SAMPLE INPUT:

```
5
-1
10 -2
10 -7 10
-6 -4 -5 10
```

SAMPLE OUTPUT:

```
5
```

Initially, the graph contains edges $(1, 2), (2, 3), (2, 4), (1, 5), (2, 5), (3, 5)$. Edge $(3, 5)$ can be removed for a cost of 5.

SAMPLE INPUT:

```
4
-1
-2 300
4 -5 6
```

SAMPLE OUTPUT:

```
9
```

Initially, the graph contains edges (1, 2), (1, 3), (2, 4). Edge (2, 4) can be removed and edge (1, 4) can be added for a total cost of $5 + 4 = 9$.

SCORING:

- Inputs 4-9: All $a_{i,j} > 0$
- Inputs 10-16: $N \leq 50$
- Inputs 17-23: No additional constraints.

Problem credits: Benjamin Qi

Contest has ended. No further submissions allowed.