Frisbee

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

UNSW's Ultimate Frisbee club has come up with a new training drill. A group of n players stand in a circle, throwing a frisbee to each other. Initially, player 1 has the frisbee.

Each player is training for two specific throws, so they are only allowed to throw the frisbee to either of two specified distinct players (perhaps including themselves).

The coach has spotted a flaw in the drill. If some player can never receive the frisbee, they will get bored. Determine whether there is any chain of throws that eventually causes each player to receive the frisbee at least once.

Input

The first line of input consists of a single integer n ($2 \leq n \leq 200,000$), representing the number of players.

n lines follow, each describing the throws available to one player. The ith such line consists of two space-separated integers a_i and b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$), representing the players who can receive the frisbee in one throw from player i.

Output

If there is any chain of throws that circulates the frisbee to all n players, display `Yes`. Otherwise, display `No`. Case will be ignored.

Scoring

For Subtask 1 (50 points):

- for all $1 \leq i \leq n$,
  - $b_i = i$.

For Subtask 2 (50 points):

- no restrictions.

Examples

standard input	standard output

4
3 1
4 2
4 3
2 4
	Yes

4
2 1
3 2
1 3
1 4
	No

6
1 3
2 6
2 5
1 4
4 5
2 6
	Yes

5
2 4
2 3
2 3
4 5
4 5
	No

Note

In the first sample case, the frisbee can reach all players: 1 — 3 — 4 — 2.

In the second sample case, player 4 will not get the frisbee.

In the third sample case, the frisbee can reach all players: 1 — 3 — 5 — 4 — 1 — 3 — 2 — 6.

In the fourth sample case, either players 2 and 3 or players 4 and 5 will not get the frisbee.

Note that only the first two sample cases comply with the constraints of Subtask 1.