Problem A. Big Numbers

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Problems with big numbers taken modulo some small number discriminate Python. © Some blue guy on Codeforces.

You are given a directed rooted tree. All edges are directed away from the root. Each edge has length which is a power of 2. The root of the tree has number 1.

Let's define two terms, which will depend on each other — a **trip** from vertex v, and a **journey** to vertex v.

A **journey** to some vertex v always starts from its parent p (it means that a journey to the root of the tree may never occur) and consists of three steps:

1. Traverse p-v edge (the edge from p to v).
2. Make a trip from v.
3. Teleport from vertex v to vertex p without traversing any edges.

A **trip** from vertex v is the following procedure:

1. Make a journey to all of the children of v (if any).
2. If v has at least one child, make a journey to some child of v one more time.

Note that any trip or journey always starts and ends at the same vertex.

Length of the trip is the total length of all traversed edges with multiplicity during the trip.

What is the maximum length of a trip from the root? Calculate it modulo 998 244 353.

Input

The first line of input contains a single integer n ($2 \le n \le 10^5$), the number of vertices in the tree.

Next $n - 1$ lines describe the tree. i-th of these lines contains two integers p_i and c_i ($1 \le p_i \le i, 0 \le c_i \le 10^{18}$) describing an edge between vertices p_i and $i + 1$ with length 2^{c_i}.

Output

Output a single integer, the maximum length of a trip from the root modulo 998 244 353.

Examples

standard input	standard output
7 1 1 2 1 1 4 2 2 1 2 5 0	52
3 1 28 1 1000000000000000000	752834992