Lab Evaluation 1

Consider a variant of singly-linked list. In this linked list, instead of pointing towards a single linked-list node, some nodes of the linked-list are pointing towards two nodes. Each node in this linked-list has two pointers named: **next** and **child**

The “child” pointer of a node may again point towards a strange singly linked list.

You need to restructure this linked-list into a singly linked list. While doing so, she must follow the following rules.

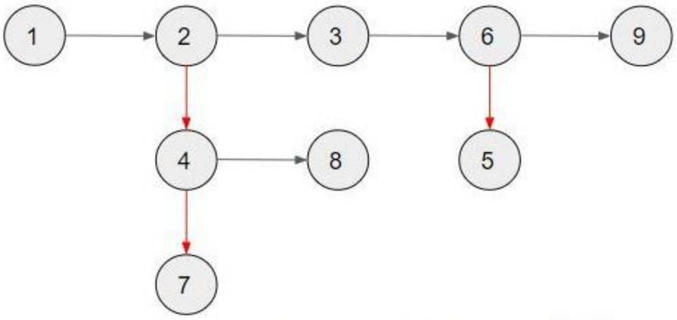
Let **cur** be a node in final restructured linked-list, then **cur.next** must occur after **cur** in the restructured linked-list.

All the nodes (if any) which are part of **cur.child** linked list must occur after the node **cur** and before the node **cur.next** in the restructured linked list.

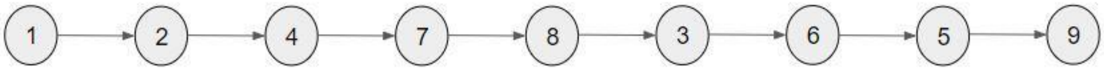
Let **cur** be a node in the restructured linked-list, then **cur.child** must be “null”.

NOTE: Refer to sample case for a visual depiction.

Input: Black Arrows=Next pointer and Red Arrows=Child Pointer



Output:



Input Format:

The first line of the input contains a single integer T - the number of test cases. The description of T test cases follows.

The first line of each test case contains a single integer: N

The following N−1 lines contain three space separated integers u,v and type. If the value of type is equal to 0, then the next pointer of node with value equal to u points towards the node with value equal to v. If the value of type is equal to 1, then the child pointer of node with value equal to u points towards the node with value equal to v.

Output Format

For each test case, the function you complete should return the restructured linked-list.

Constraints

$1 \le T \le 100$

$1 \le N \le 2 * 10^5$

It is guaranteed that the sum of N over all test cases is less than or equal to $3 * 10^5$

Sample Input:

```
1
9
4 7 1
1 2 0
2 3 0
6 9 0
3 6 0
2 4 1
4 8 0
6 5 1
```

Output:

```
1 2 4 7 8 3 6 5 9
```