# Grid Tree

A rooted tree consisting of $N$ nodes numbered from $0$ to $N-1$ is given. Node $0$ is the root of this tree, and every node has either $0$ or $2$ children. Nodes with exactly two children have designated left and right children. Each edge $e$ of the tree has a positive integer length $c_e$.

We will draw this tree on a 2D coordinate plane. Each node $v$ is drawn at a distinct lattice point $m_v = (x_v, y_v)$. Here, the root node must be drawn at $m_0 = (0, 0)$. A lattice point means a point where both the $x$ and $y$ coordinates are integers.

The edge $e = (p, v)$ connecting $v$ and its parent $p$ is drawn as a path connecting $m_p$ and $m_v$ on the coordinate plane. This path must satisfy all of the following conditions:

- The direction of movement along the path from $m_p$ towards $m_v$ must always be in the positive x-direction or positive y-direction. Specifically, moving in a direction where both the $x$ and $y$ coordinates increase is impossible. Also, directions can only be changed at lattice points. That is, if the length of the path is $k$, its direction can be changed only at exactly $k-1$ points.
- If $v$ is the left child of $p$, the path must start from $m_p$ in the positive x-direction.
- If $v$ is the right child of $p$, the path must start from $m_p$ in the positive y-direction.
- The length of the path must be at least the corresponding edge length $c_e$.
- Paths **must not intersect**: In other words, an interior point of any path (a point other than the start and end points) must not be contained in any other path.

Let's define the **depth** of node $v$ in the tree drawing as $L(v) = x_v + y_v$. In the tree we draw, all nodes with no children must have the same depth. Let's call this depth the **grid depth**.

Find the minimum **grid depth** for all valid tree drawings.

## Implementation Details

You need to implement the following function.

```
long long compute_min_depth(int N, vector<int> P, vector <int> C, vector<int>
D)
```

- $N$: The number of nodes.
- $P, C, D$: Integer arrays of size $N-1$. For all $1 \leq i \leq N-1$, the parent of node $i$ is node $P[i-1]$. Let $e$ be the edge connecting node $i$ and its parent; then $c_e = C[i-1]$. If $D[i-1] = 0$, node $i$ is the left child of its parent, and if $D[i-1] = 1$, node $i$ is the right child of its parent.
- It can be proved that a tree drawing satisfying the conditions always exists. This function should return the minimum value of the grid depth among them.
- This function is called exactly once.

Your submitted source code must not execute any input/output functions.

## Constraints

- The given edges form a tree rooted at node $0$.
- The number of children for each node is either $0$ or $2$.
- $3 \leq N \leq 200\,000$
- For all $i$, $0 \leq P[i] \leq N-1$ $(0 \leq i \leq N-2)$
- For all $i$, $1 \leq C[i] \leq 10^9$ $(0 \leq i \leq N-2)$
- For all $i$, $0 \leq D[i] \leq 1$ $(0 \leq i \leq N-2)$

## Subtasks

Let the distance between two nodes in the tree be defined as the sum of the lengths of the edges constituting the unique path connecting the two nodes.

| Subtask | Score | Additional Constraints |
|---|---|---|
| 1 | 10 | $N \leq 7$ |
| 2 | 8 | For every node $v$ with exactly $2$ children, one of the children of $v$ has $0$ children. |
| 3 | 21 | $N \leq 5000$, for all nodes $v$ with $0$ children, the distance between node $0$ and $v$ is the same value $K$ $(\leq 2500)$. |
| 4 | 29 | $N \leq 5000$, for all nodes $v$, the distance between node $0$ and $v$ is at most $2500$. |
| 5 | 32 | No additional constraints. |

## Scoring

For Subtask $3$, if a tree drawing with grid depth exactly $K$ does not exist, it is accepted as correct if `compute_min_depth` returns $-1$. More precisely:

- Test cases where a tree drawing with grid depth $K$ **exists**:

    - If $K$ is returned, full score is awarded.
    - Otherwise, 0 points are awarded.
- Test cases where a tree drawing with grid depth $K$ **does not exist**:
    - If the minimum grid depth is returned, full score is awarded.
    - If $-1$ is returned, full score is awarded.
    - Otherwise, 0 points are awarded.

Note that in Subtask $3$, for all nodes $v$ with $0$ children, the distance between node $0$ and $v$ is the same, and that distance is $K$.

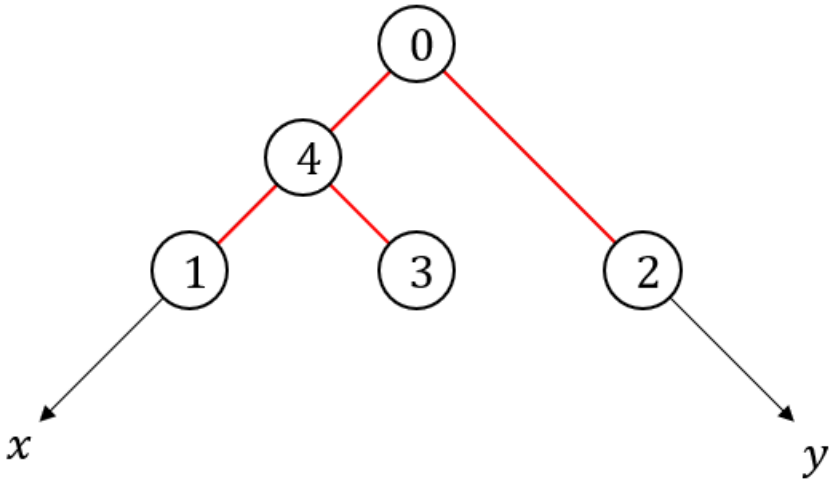
## Examples

Example 1

Consider the following call:

```
compute_min_depth(5, [4, 0, 4, 0], [1, 2, 1, 1], [0, 1, 1, 0])
```

- We can draw a tree with a grid depth of $2$ as shown in the figure below.




It can be proved that there is no tree drawing with a grid depth less than $2$.

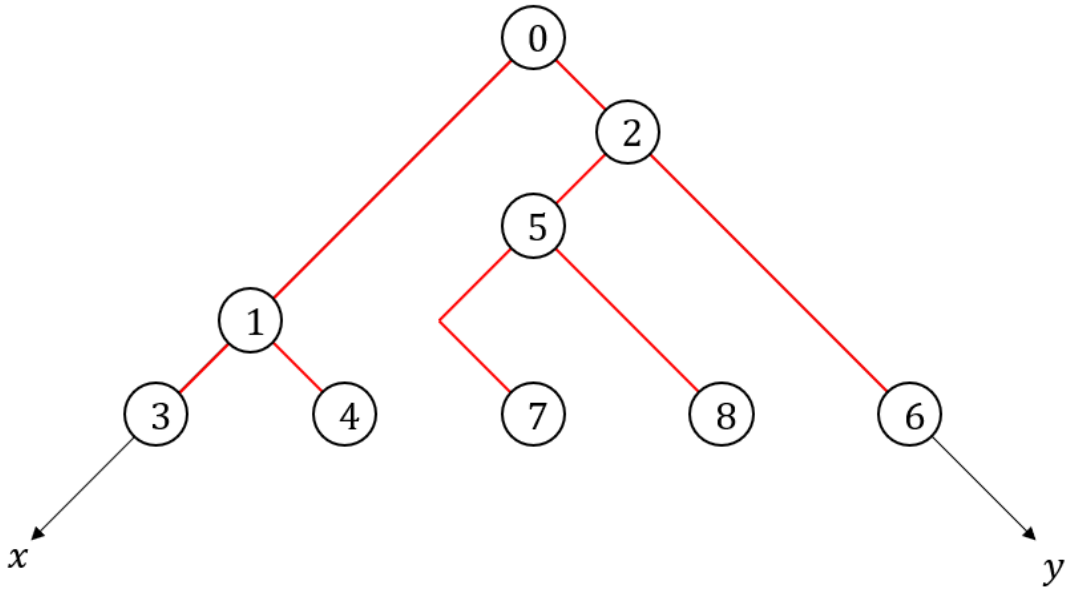
Therefore, the function should return $2$.

Example 2

Consider the following call:

```
compute_min_depth(9, [0, 0, 1, 1, 2, 2, 5, 5], [2, 1, 1, 1, 1, 1, 1, 1], [0,
1, 0, 1, 0, 1, 0, 1])
```

- We can draw a tree with a grid depth of $4$ as shown in the figure below.




Therefore, the function should return $4$.

## Sample Grader

The input format for the sample grader is as follows.

- line $1$: $N$
- For all $0 \leq i < N - 1$:
  - line $2 + i$: $P[i]\ C[i]\ D[i]$

The sample grader prints the answer in the following format:

- line $1$: The return value of `compute_min_depth`