

AIZU GRL 5A

Diameter of a Tree

Prof. Edson Alves

Faculdade UnB Gama

Given a tree T with non-negative weight, find the diameter of the tree.

The diameter of a tree is the maximum distance between two nodes in a tree.

Dada uma árvore T com pesos não-negativos, encontre o diâmetro desta árvore.

O diâmetro de uma árvore é a distância máxima entre dois vértices de uma árvore.

Input

n
 $s_1 \quad t_1 \quad w_1$
 $s_2 \quad t_2 \quad w_2$
 \dots
 $s_{n-1} \quad t_{n-1} \quad w_{n-1}$

The first line consists of an integer n which represents the number of nodes in the tree. Every node has a unique ID from 0 to $n - 1$ respectively.

In the following $n - 1$ lines, edges of the tree are given. s_i and t_i represent end-points of the i -th edge (undirected) and w_i represents the weight (distance) of the i -th edge.

Entrada

n		
s_1	t_1	w_1
s_2	t_2	w_2
\dots		
s_{n-1}	t_{n-1}	w_{n-1}

A primeira linha consiste em um inteiro n , o qual representa o número de nós na árvore. Cada nó tem um identificador único entre 0 e $n - 1$, respectivamente.

Nas próximas $n - 1$ linhas são dadas as arestas da árvore. s_i e t_i representam os pontos terminais da i -ésima aresta (não-direcionada) e w_i representa o peso (distância) da i -ésima aresta.

Output

Print the diameter of the tree in a line.

Constraints

- ▶ $1 \leq n \leq 100,000$
- ▶ $0 \leq w_i \leq 1,000$

Saída

Imprima, em uma linha, o diâmetro da árvore.

Restrições

- ▶ $1 \leq n \leq 100,000$
- ▶ $0 \leq w_i \leq 1,000$

Exemplo de entrada e saída

Exemplo de entrada e saída

4

Exemplo de entrada e saída

4
↑
de nós

Exemplo de entrada e saída

4

1

0

2

3

Exemplo de entrada e saída

4

0 1 2

0

1

2

3

Exemplo de entrada e saída

4
0 1 2
↑
 u

0

1

2

3

Exemplo de entrada e saída

4
0 1 2
↑
 v

1

0

2

3

Exemplo de entrada e saída

4
0 1 2
↑
 w

1

0

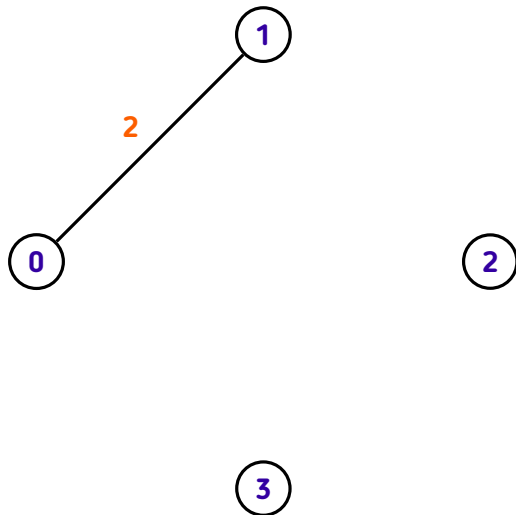
2

3

Exemplo de entrada e saída

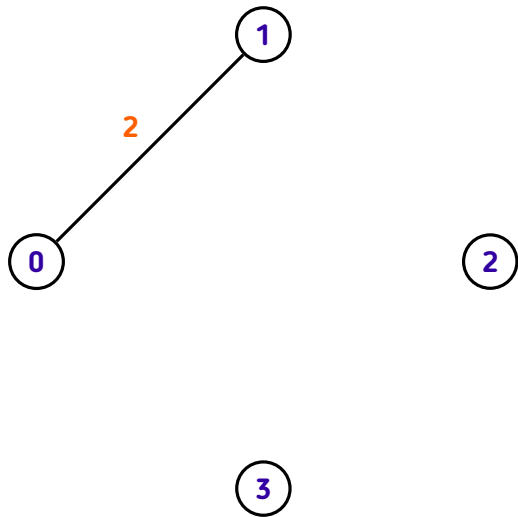
4

0 1 2



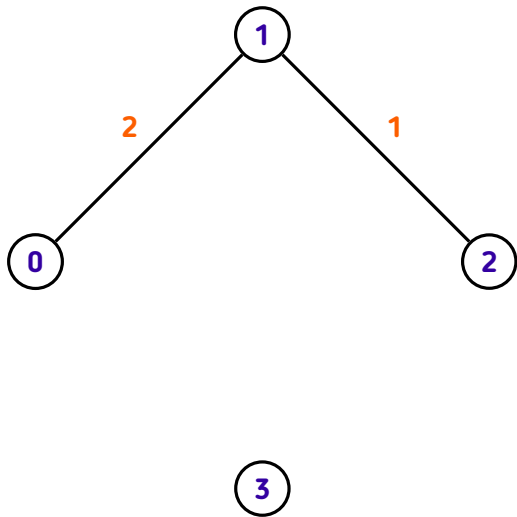
Exemplo de entrada e saída

4
0 1 2
1 2 1



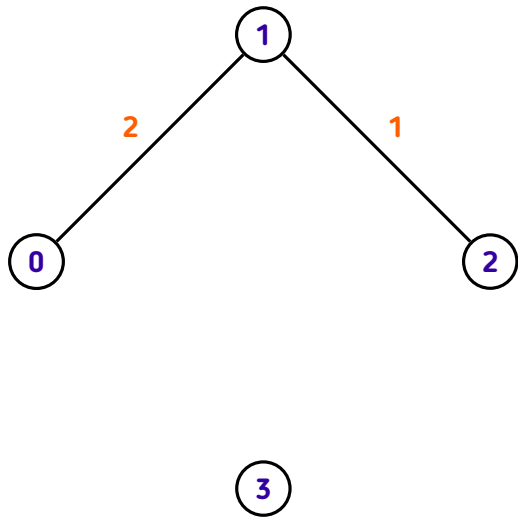
Exemplo de entrada e saída

4
0 1 2
1 2 1



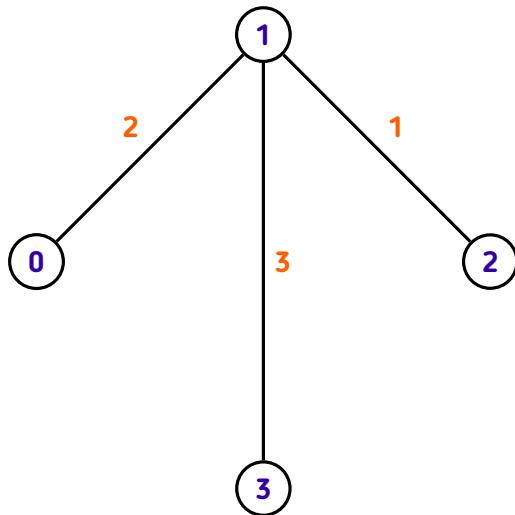
Exemplo de entrada e saída

4
0 1 2
1 2 1
1 3 3



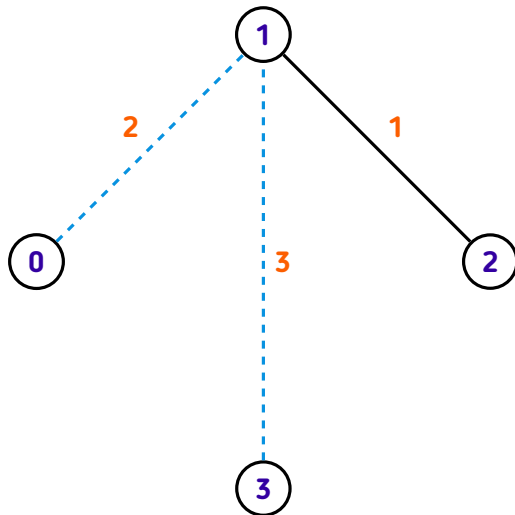
Exemplo de entrada e saída

4
0 1 2
1 2 1
1 3 3



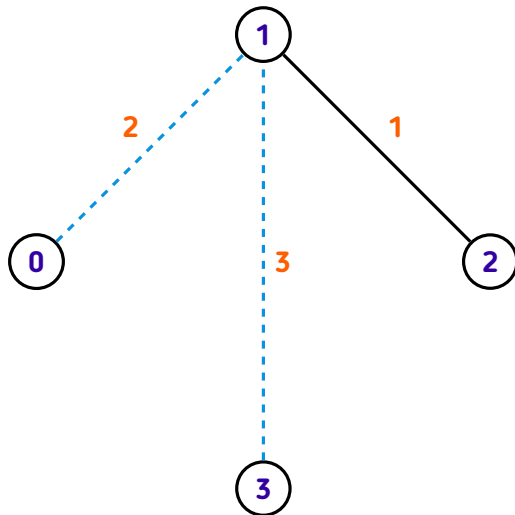
Exemplo de entrada e saída

4
0 1 2
1 2 1
1 3 3



Exemplo de entrada e saída

4
0 1 2
1 2 1
1 3 3
↓
 $2 + 3 = 5$



Exemplo de entrada e saída

Exemplo de entrada e saída

4

Exemplo de entrada e saída

4

1

0

2

3

Exemplo de entrada e saída

4

0 1 1

0

1

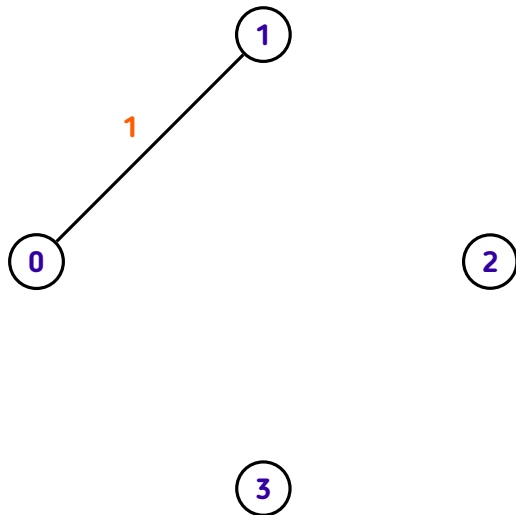
2

3

Exemplo de entrada e saída

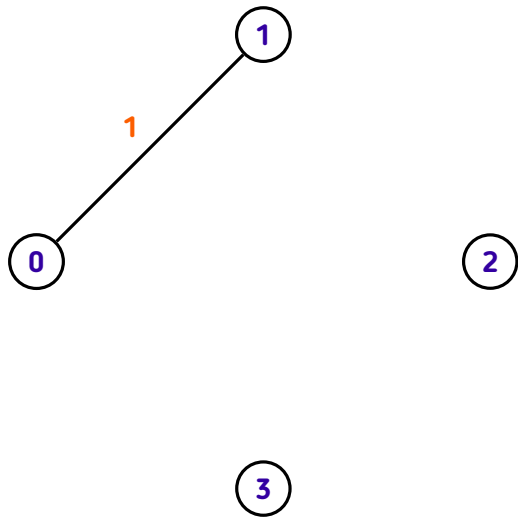
4

0 1 1



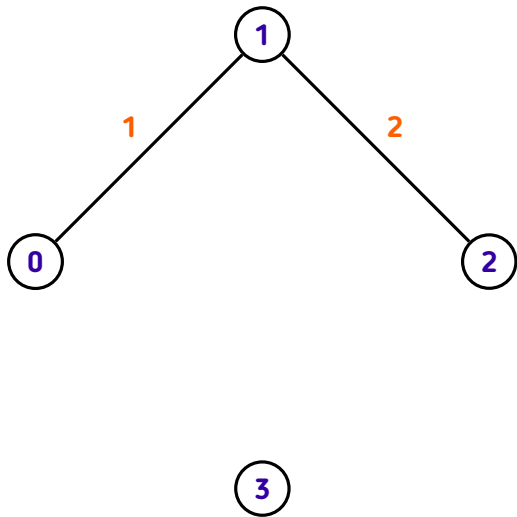
Exemplo de entrada e saída

4
0 1 1
1 2 2



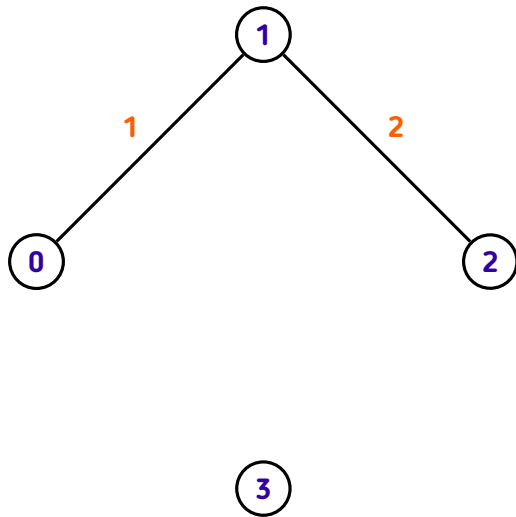
Exemplo de entrada e saída

4
0 1 1
1 2 2



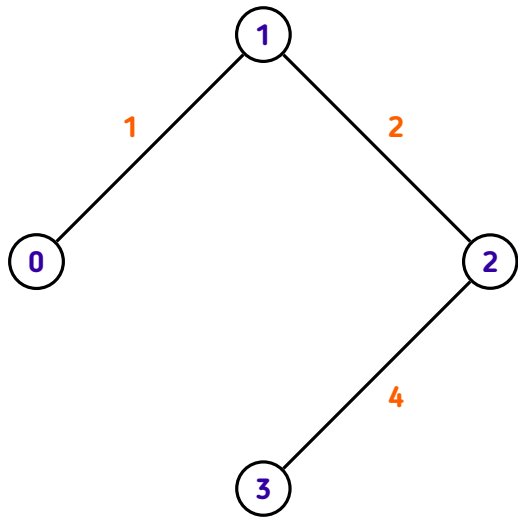
Exemplo de entrada e saída

4
0 1 1
1 2 2
2 3 4



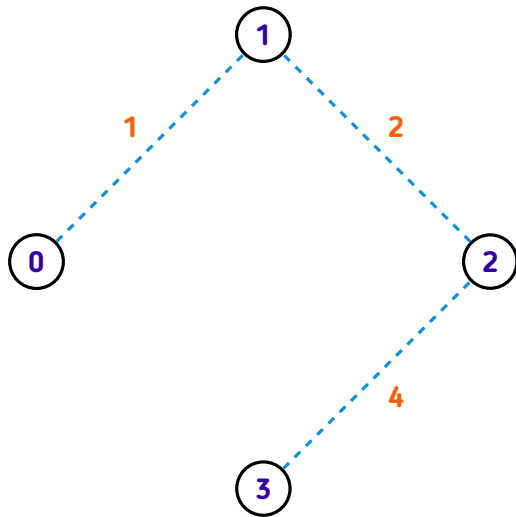
Exemplo de entrada e saída

4
0 1 1
1 2 2
2 3 4



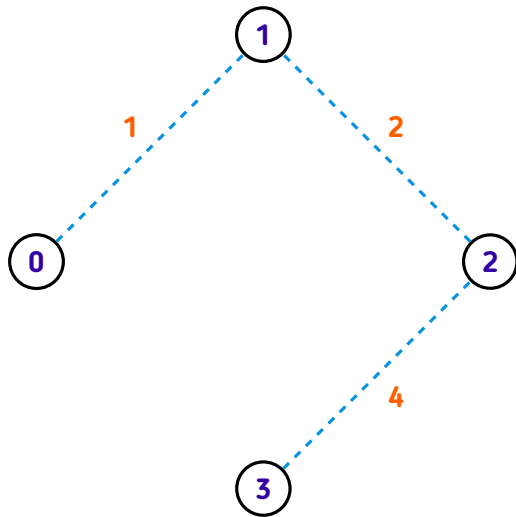
Exemplo de entrada e saída

4
0 1 1
1 2 2
2 3 4



Exemplo de entrada e saída

$$\begin{array}{cccc} & & 4 & \\ & 0 & 1 & 1 \\ & 1 & 2 & 2 \\ & 2 & 3 & 4 \\ & \downarrow & & \\ 1 + 2 + 4 = 7 \end{array}$$



Solução

Solução

- ★ O diâmetro também pode ser computado por meio de duas DFS

Solução

- ★ O diâmetro também pode ser computado por meio de duas DFS
- ★ Isto porque, em árvores, todos caminhos mínimos são únicos

Solução

- ★ O diâmetro também pode ser computado por meio de duas DFS
- ★ Isto porque, em árvores, todos caminhos mínimos são únicos
- ★ A DFS encurta o código

Solução

- ★ O diâmetro também pode ser computado por meio de duas DFS
- ★ Isto porque, em árvores, todos caminhos mínimos são únicos
- ★ A DFS encurta o código
- ★ A função `max_element()` do C++ retorna um iterador para o elemento máximo no intervalo passado como parâmetro

```
void dfs(int u, int p, vector<int>& dist)
{
    for (auto [v, w] : adj[u])
    {
        if (v != p)
        {
            dist[v] = dist[u] + w;
            dfs(v, u, dist);
        }
    }
}
```

```
int solve(int n)
{
    vector<int> dist(n + 1, 0);

    dfs(0, -1, dist);

    auto v = (int) (max_element(dist.begin(), dist.end()) - dist.begin());

    dist[v] = 0;

    dfs(v, -1, dist);

    return *max_element(dist.begin(), dist.end());
}
```