AtCoder Beginner Contest 126

Problem D – Even Relation

Prof. Edson Alves

Faculdade UnB Gama

We have a tree with N vertices numbered 1 to N. The i-th edge in the tree connects Vertex u_i and Vertex v_i, and its length is w_i. Your objective is to paint each vertex in the tree white or black (it is fine to paint all vertices the same color) so that the following condition is satisfied:

- *For any two vertices painted in the same color, the distance between them is an even number.*

Find a coloring of the vertices that satisfies the condition and print it. It can be proved that at least one such coloring exists under the constraints of this problem.

Nós temos uma árvore com N vértices numerados de 1 a N. A i-ésima aresta da árvore conecta o Vértice u_i e o Vértice v_i, e seu comprimento é w_i. Seu objetivo é pintar cada vértice da árvore de branco ou preto (é válido pintar todos os vértices com uma mesma cor) de modo que a seguinte condição é satisfeita:

- Para quaisquer dois vértices pintados com a mesma cor, a distância entre eles é um número par.

Determine uma coloração dos vértices que satisfaça a condição e a imprima. Pode ser provado que existe no mínimo uma coloração que atenda a condição imposta pelo problema.

Constraints

- ***All values in input are integers.***
- $1 \leq N \leq 10^5$
- $1 \leq u_i < v_i \leq N$
- $1 \leq w_i \leq 10^9$

Restrições

- **Todos os valores da entrada são inteiros.**
- $1 \leq N \leq 10^5$
- $1 \leq u_i < v_i \leq N$
- $1 \leq w_i \leq 10^9$

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{ccc}
N & & \\
u_1 & v_1 & w_1 \\
u_2 & v_2 & w_2 \\
\vdots & & \\
u_{N-1} & v_{N-1} & w_{N-1}
\end{array}
$$

Output

Print a coloring of the vertices that satisfies the condition, in N lines. The i-th line should contain `0` if Vertex i is painted white and `1` if it is painted black.

If there are multiple colorings that satisfy the condition, any of them will be accepted.

Entrada

A entrada é dada na Entrada Padrão no seguinte formato:

$$
\begin{array}{ccc}
N & & \\
u_1 & v_1 & w_1 \\
u_2 & v_2 & w_2 \\
\vdots & & \\
u_{N-1} & v_{N-1} & w_{N-1}
\end{array}
$$

Saída

Imprima uma coloração dos vértices que satisfaça a condição, em N linhas. A i-ésima linha deve conter `0` se o Vértice i deve ser pintado branco ou `1` se ele deve ser pintado preto.

Se há múltiplas colorações que satisfaçam a condição, qualquer uma delas será aceita.

Exemplo de entrada e saída

Exemplo de entrada e saída

3

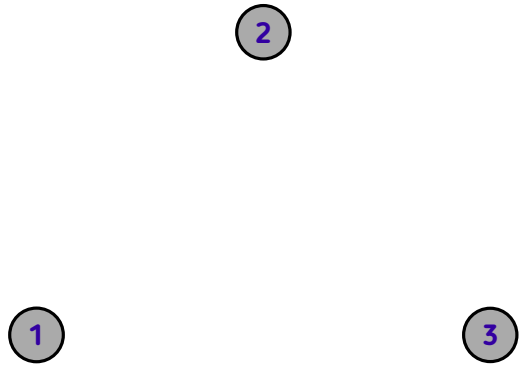
Exemplo de entrada e saída

3

 # número de vértices

Exemplo de entrada e saída

3



Exemplo de entrada e saída

```
3
1 2 2
```

Exemplo de entrada e saída

3
1 2 2

2

1

3

Exemplo de entrada e saída

3
1 2 2

Exemplo de entrada e saída

3
1 2 2

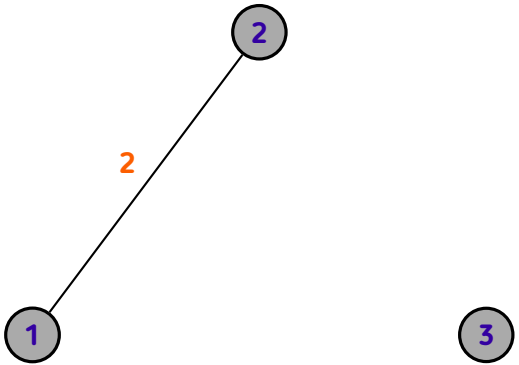
2

1

3

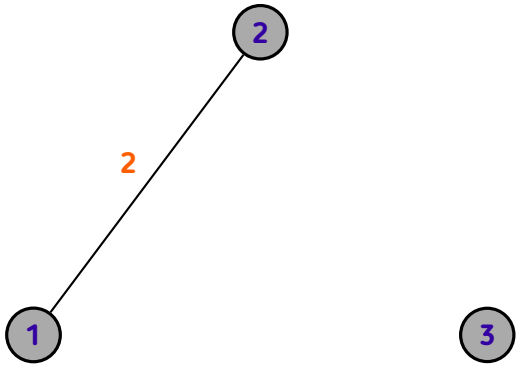
Exemplo de entrada e saída

```
3
1 2 2
```



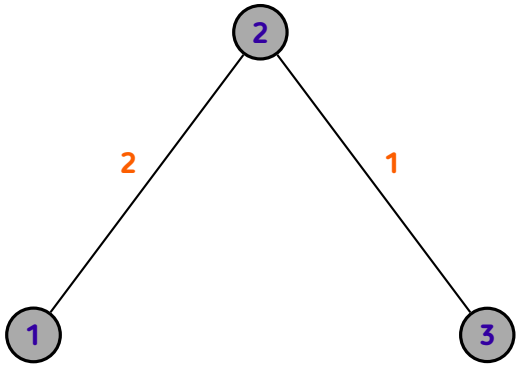
Exemplo de entrada e saída

```
3
1 2 2
2 3 1
```



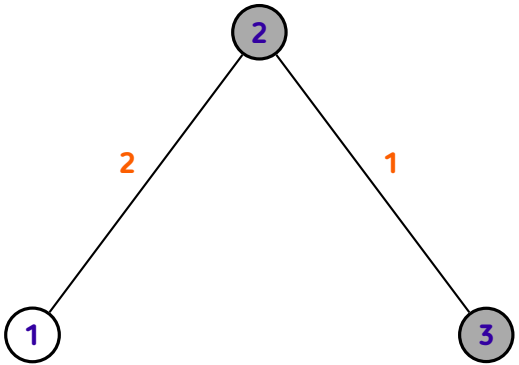
Exemplo de entrada e saída

```
3
1 2 2
2 3 1
```



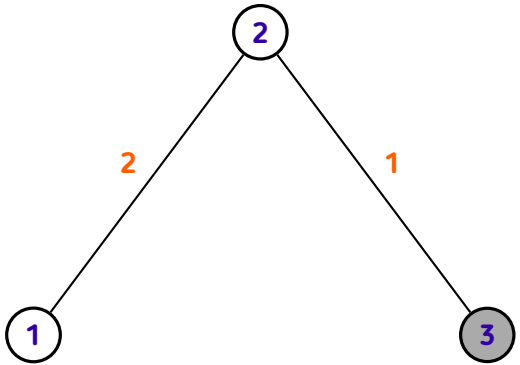
Exemplo de entrada e saída

```
3
1 2 2
2 3 1
```



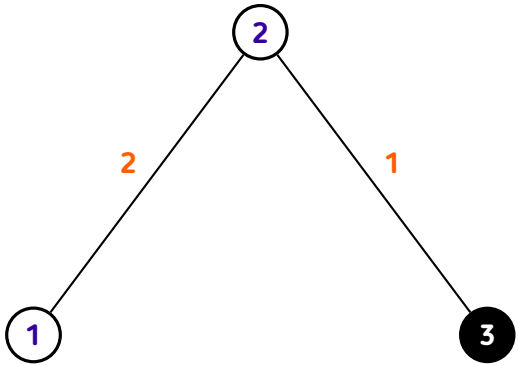
Exemplo de entrada e saída

```
3
1 2 2
2 3 1
```



Exemplo de entrada e saída

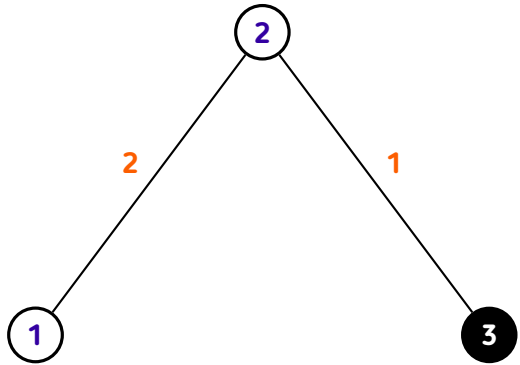
```
3
1 2 2
2 3 1
```



Exemplo de entrada e saída

```
3
1 2 2
2 3 1
```

```
0 0 1
```



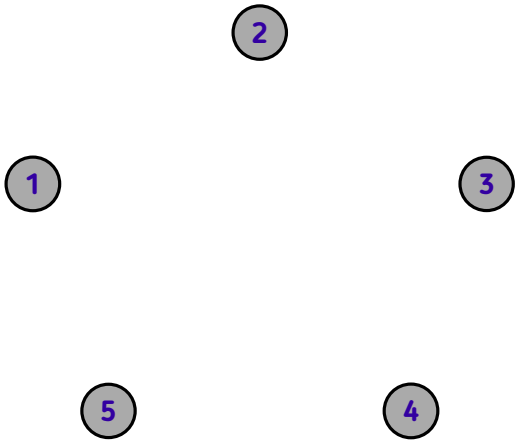
Exemplo de entrada e saída

Exemplo de entrada e saída

5

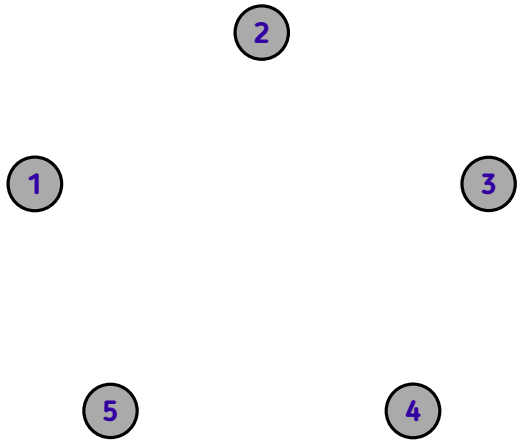
Exemplo de entrada e saída

5



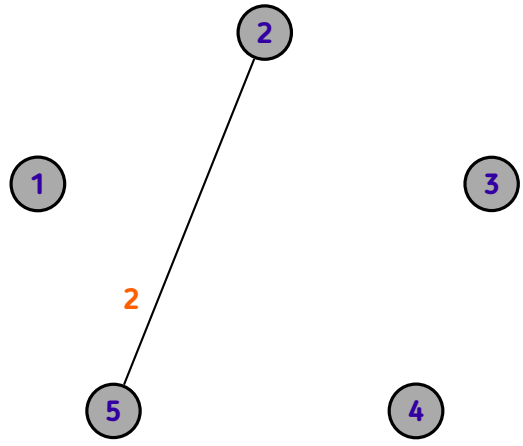
Exemplo de entrada e saída

```
5
2 5 2
```



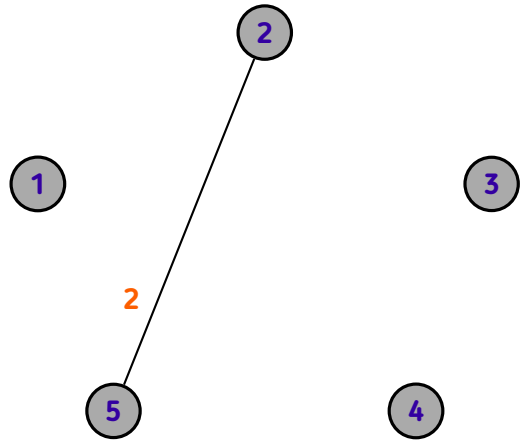
Exemplo de entrada e saída

5
2 5 2



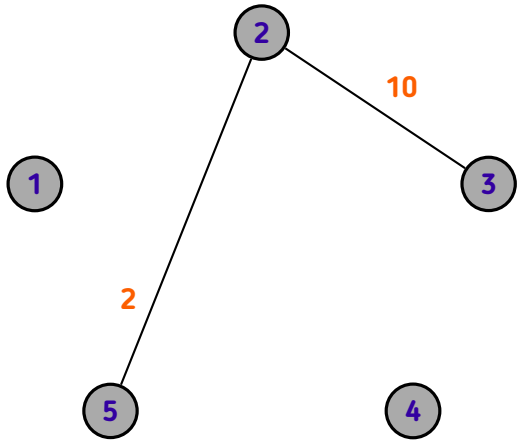
Exemplo de entrada e saída

```
5
2 5 2
2 3 10
```



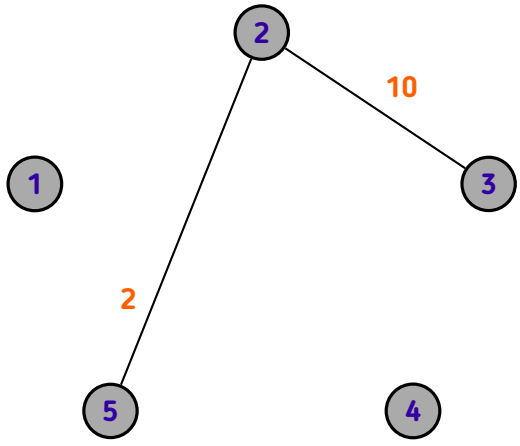
Exemplo de entrada e saída

```
5
2 5 2
2 3 10
```



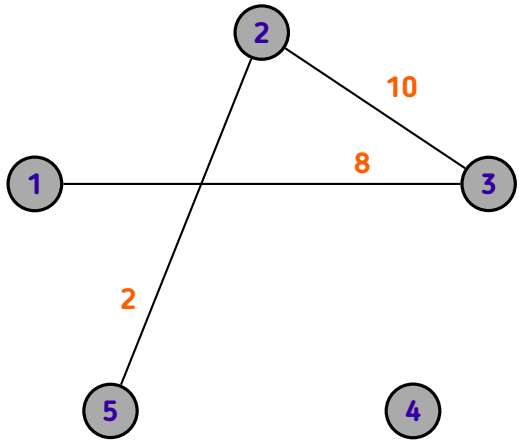
Exemplo de entrada e saída

```
5
2 5 2
2 3 10
1 3 8
```



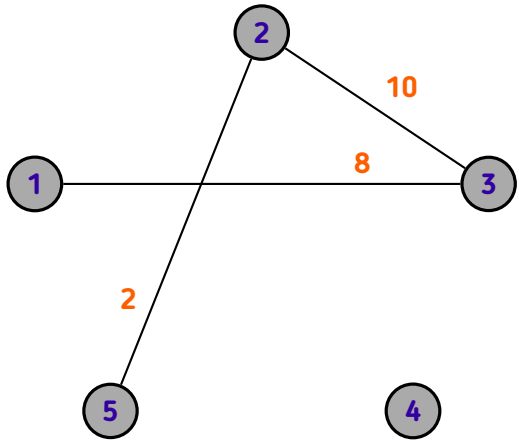
Exemplo de entrada e saída

```
5
2 5 2
2 3 10
1 3 8
```



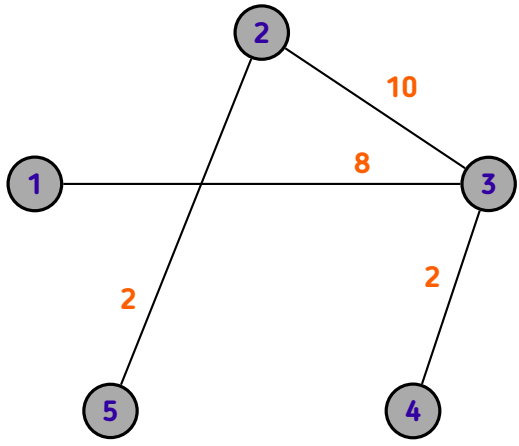
Exemplo de entrada e saída

```
5
2 5 2
2 3 10
1 3 8
3 4 2
```



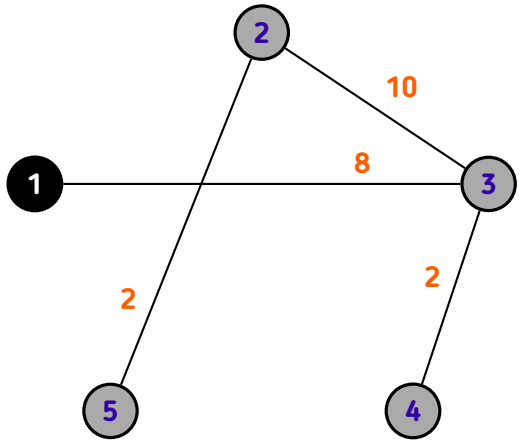
Exemplo de entrada e saída

```
5
2 5 2
2 3 10
1 3 8
3 4 2
```



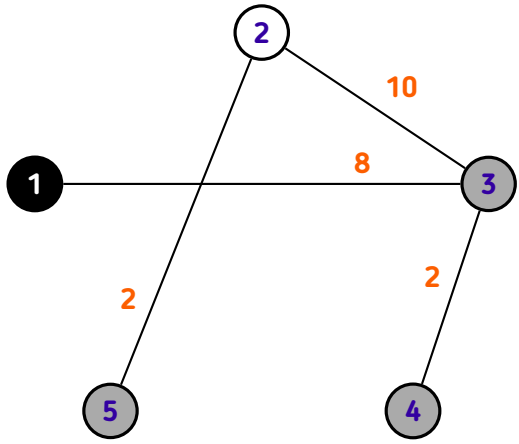
Exemplo de entrada e saída

```
5
2 5 2
2 3 10
1 3 8
3 4 2
```



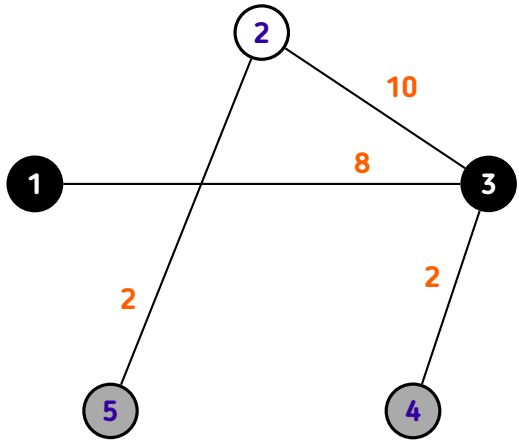
Exemplo de entrada e saída

```
5
2 5 2
2 3 10
1 3 8
3 4 2
```



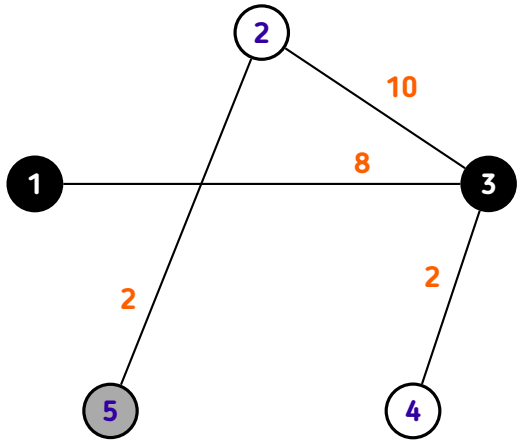
Exemplo de entrada e saída

```
5
2 5 2
2 3 10
1 3 8
3 4 2
```



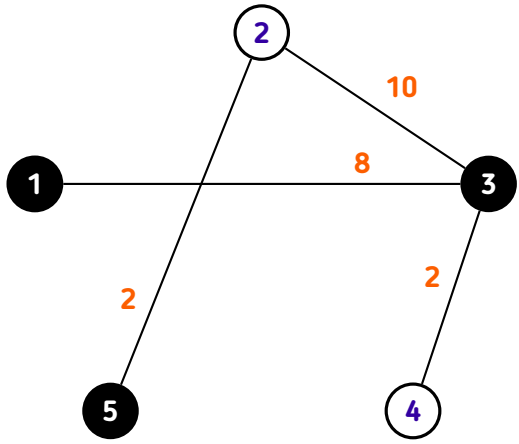
Exemplo de entrada e saída

```
5
2 5 2
2 3 10
1 3 8
3 4 2
```



Exemplo de entrada e saída

```
5
2 5 2
2 3 10
1 3 8
3 4 2
```

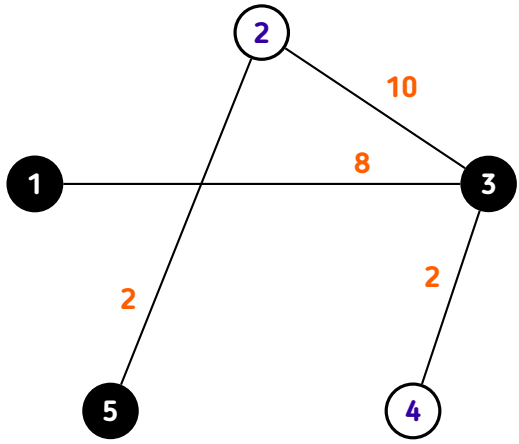


Exemplo de entrada e saída

```
5
2 5 2
2 3 10
1 3 8
3 4 2
```

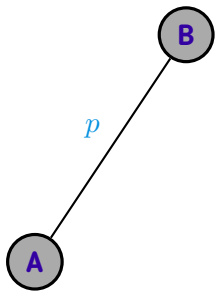
↓

```
1 0 1 0 1
```



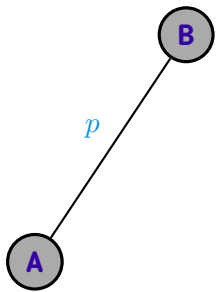
Solução

Solução



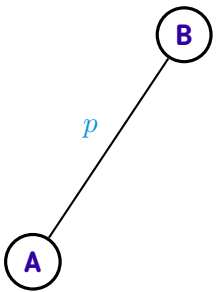
Solução

Se o peso w é um número par p, ambos vértices devem ter a mesma cor

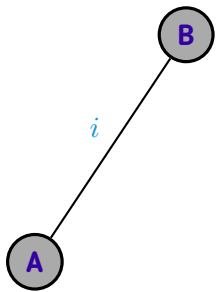


Solução

Se o peso w é um número par p, ambos vértices devem ter a mesma cor

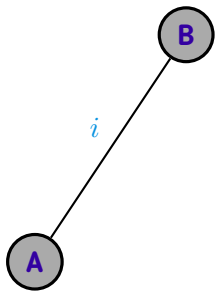


Solução



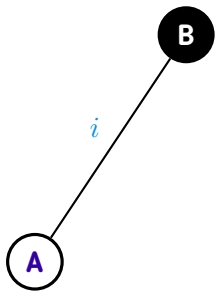
Solução

Se o peso w é um número ímpar i, ambos vértices devem ter cores distintas



Solução

Se o peso w é um número ímpar i, ambos vértices devem ter cores distintas



Solução

Estes dois critérios são suficientes para a resolução do problema!

Solução

Estes dois critérios são suficientes para a resolução do problema!

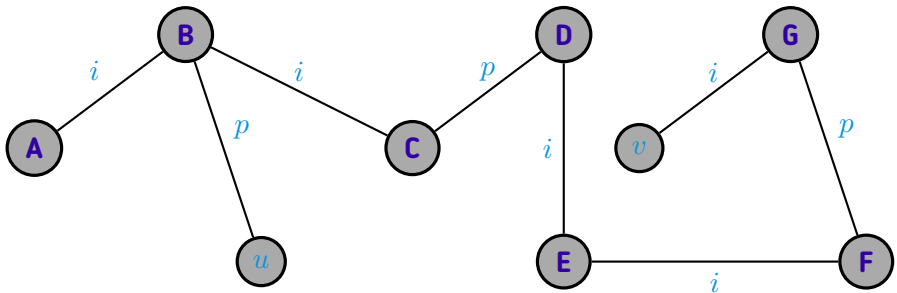
Isto porque cada aresta com peso ímpar em um caminho corresponde a uma troca de cores!

Solução

Se a distância de u a v é par, ou todas arestas são pares, ou há um número par de arestas ímpares no caminho de u a v.

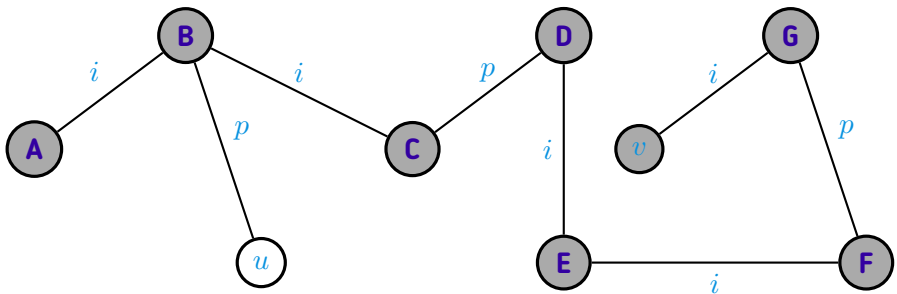
Solução

Se a distância de u a v é par, ou todas arestas são pares, ou há um número par de arestas ímpares no caminho de u a v.



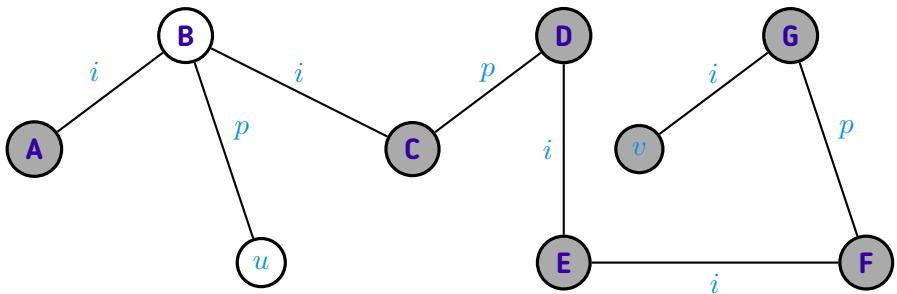
Solução

Se a distância de u a v é par, ou todas arestas são pares, ou há um número par de arestas ímpares no caminho de u a v.



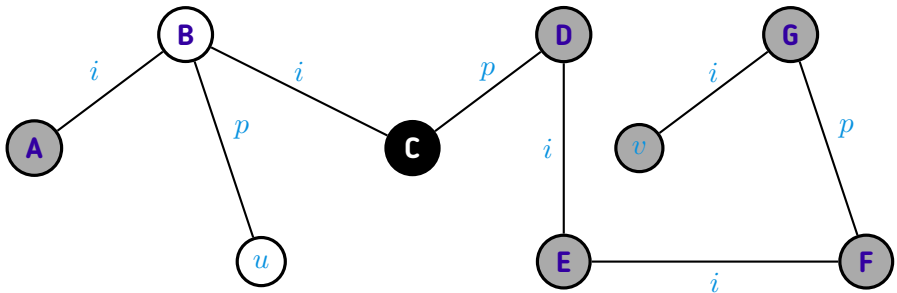
Solução

Se a distância de u a v é par, ou todas arestas são pares, ou há um número par de arestas ímpares no caminho de u a v.



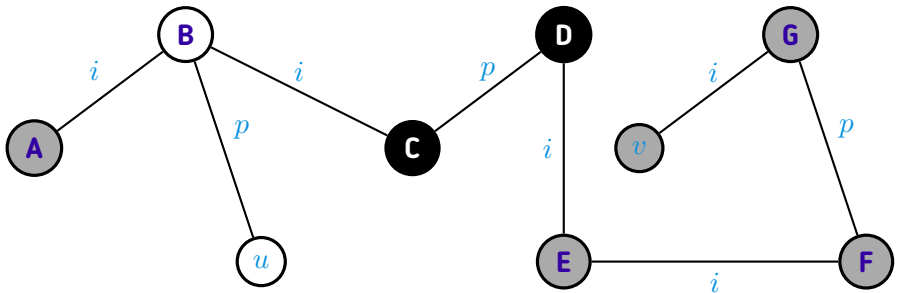
Solução

Se a distância de u a v é par, ou todas arestas são pares, ou há um número par de arestas ímpares no caminho de u a v.



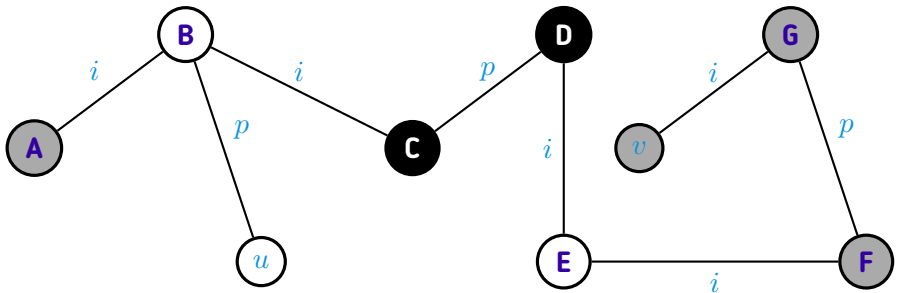
Solução

Se a distância de u a v é par, ou todas arestas são pares, ou há um número par de arestas ímpares no caminho de u a v.



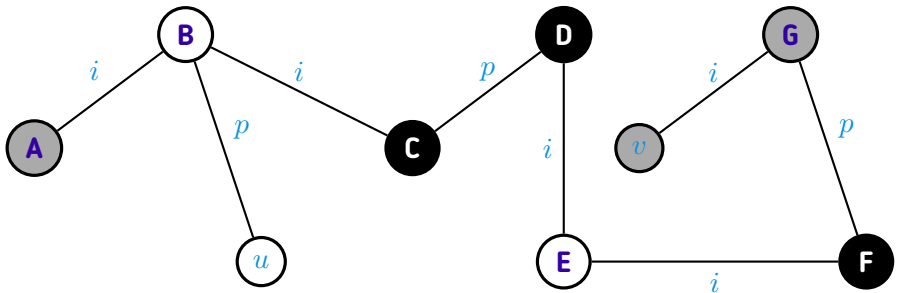
Solução

Se a distância de u a v é par, ou todas arestas são pares, ou há um número par de arestas ímpares no caminho de u a v.



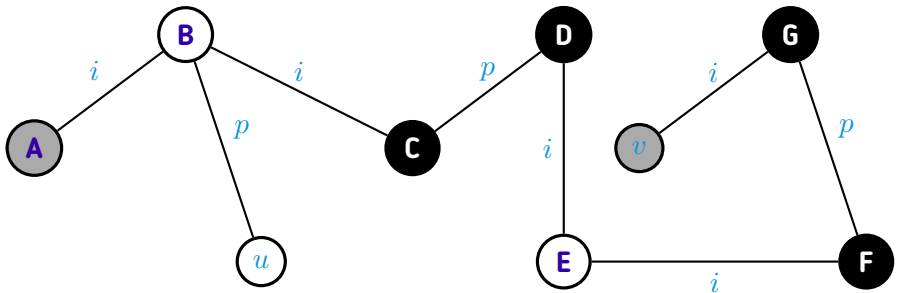
Solução

Se a distância de u a v é par, ou todas arestas são pares, ou há um número par de arestas ímpares no caminho de u a v.



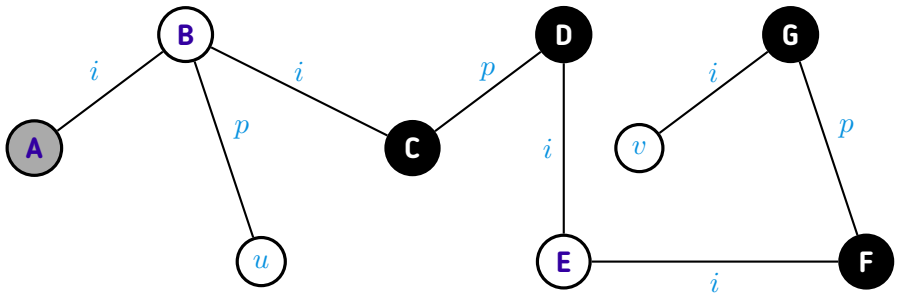
Solução

Se a distância de u a v é par, ou todas arestas são pares, ou há um número par de arestas ímpares no caminho de u a v.



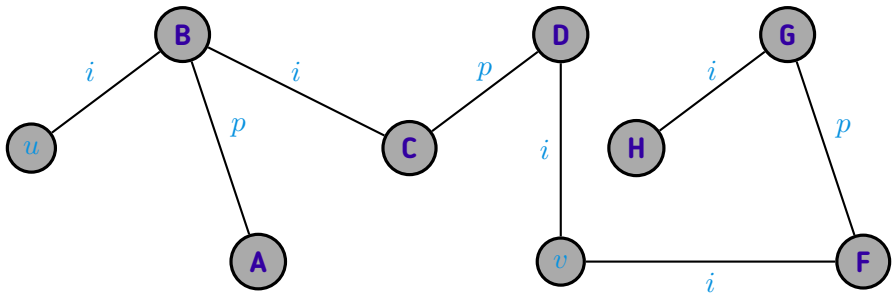
Solução

Se a distância de u a v é par, ou todas arestas são pares, ou há um número par de arestas ímpares no caminho de u a v.



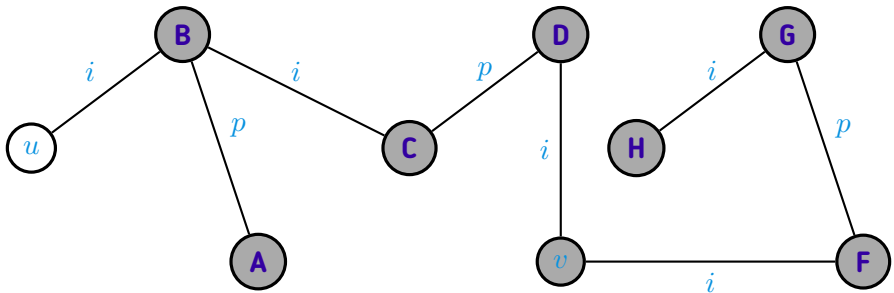
Solução

Se a distância de u a v é par, ou todas arestas são pares, ou há um número par de arestas ímpares no caminho de u a v.



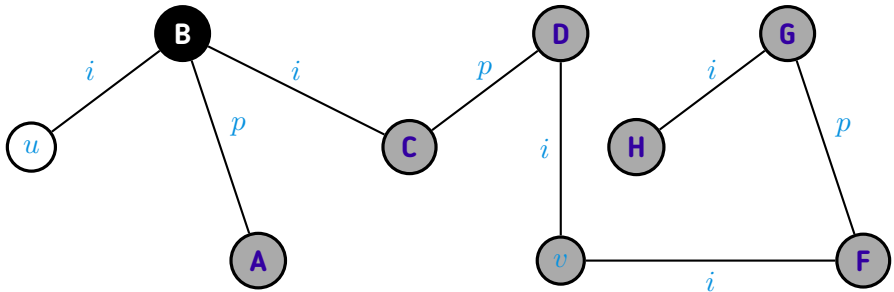
Solução

Se a distância de u a v é par, ou todas arestas são pares, ou há um número par de arestas ímpares no caminho de u a v.



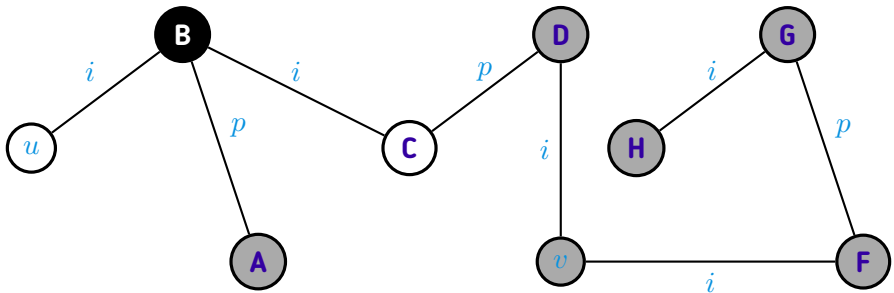
Solução

Se a distância de u a v é par, ou todas arestas são pares, ou há um número par de arestas ímpares no caminho de u a v.



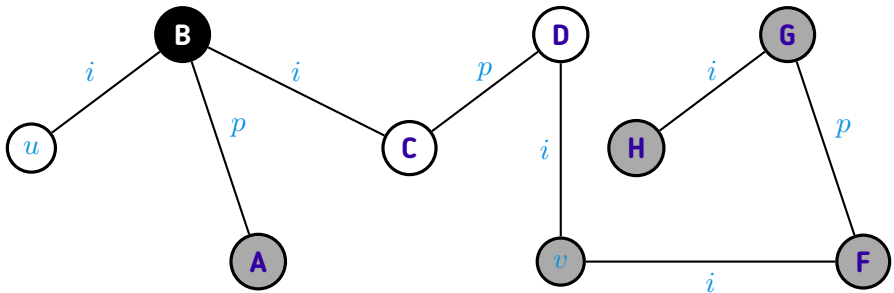
Solução

Se a distância de u a v é par, ou todas arestas são pares, ou há um número par de arestas ímpares no caminho de u a v.



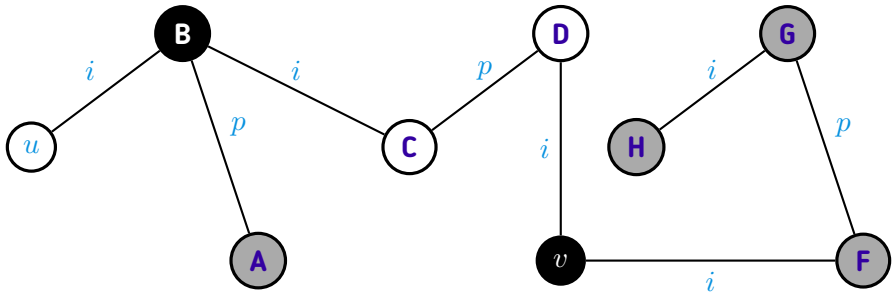
Solução

Se a distância de u a v é par, ou todas arestas são pares, ou há um número par de arestas ímpares no caminho de u a v.



Solução

Se a distância de u a v é par, ou todas arestas são pares, ou há um número par de arestas ímpares no caminho de u a v.



```cpp
void dfs(int u, int color, vector<int>& cs)
{
    if (cs[u] > -1)
        return;

    cs[u] = color;

    for (auto [v, w] : adj[u])
    {
        if (w % 2 == 0)
            dfs(v, color, cs);
        else
            dfs(v, 1 - color, cs);
    }
}
```

```cpp
vector<int> solve(int N)
{
    vector<int> cs(N + 1, -1);

    dfs(1, 0, cs);

    return cs;
}
```