

## DESAFIO: Transporte de computadores

Forma de entrega: [link do programa salvo no Gist do Github](#)

Linguagens aceitas: Javascript, Java, C#, Python

A empresa PPC (Programadores Procrastinadores Compulsivos), faz entregas de computadores de última geração aos seus clientes, localizados em todo o país.

Para manter um tratamento VIP com todos seus clientes, para cada entrega que a empresa faz, ela atribui um único entregador, que fará também a instalação da máquina. O entregador procura sempre o menor caminho possível, de modo a minimizar as distâncias percorridas.

Há uma lista de clientes interessados em comprar um dos computadores da PPC, mas o gerente não sabe qual será o custo total de enviar entregadores para todos eles. Você precisa ajudá-lo, fazendo um programa que informe qual é o custo mínimo de fazer todas essas entregas VIP, dada uma lista que informa a cidade dos clientes interessados.

Você recebe um inteiro  $n$ , indicando que existem  $n$  cidades estão numeradas de 0 a  $n - 1$ , e também uma lista `connections` onde `connections[i] = [xi, yi, costi]` indica que o custo conectando as cidades  $x_i$  e  $y_i$  (conexão bidirecional) é  $cost_i$ .

Adicionalmente, você recebe também uma lista `locations` onde `locations[i] = [li]`, indicando a cidade de rótulo  $li$  onde o cliente  $i$  reside. A empresa está sempre localizada na cidade 0.

Seu programa deverá calcular o custo total de enviar um entregador para a cidade de cada um dos clientes interessados, de forma que essa distância percorrida seja mínima.

Obs: considere que tomando uma cidade qualquer, sempre é possível atingir todas as outras (grafo conectado).

Entrada 1	Saída 1
<pre>{   "n": 6,   "connections": [     [0, 1, 1],     [0, 2, 2],     [0, 3, 3],     [1, 4, 3],     [2, 4, 2],     [3, 4, 3],     [4, 5, 4]   ],   "locations": [1, 2, 3, 4, 5] }</pre>	18

Explicação: Os menores custos para entregar nas cidades dos clientes são, respectivamente, 1, 2, 3, 4 e 8, somando esses valores obtemos 18.

Entrada 2	Saída 2
<pre>{   "n": 6,   "connections": [     [0, 1, 1],     [0, 2, 2],     [0, 3, 3],     [1, 4, 3],     [2, 4, 2],     [3, 4, 3],     [4, 5, 4]   ],   "locations": [2, 4, 5, 5] }</pre>	22

Explicação: Os menores custos para entregar nas cidades dos clientes são, respectivamente, 2, 4, 8, 8, somando esses valores obtemos 22.

Assinaturas:

Javascript:

```
function totalDeliveryCost(n, connections, locations)
```

Java:

```
public String[] totalDeliveryCost(int n, int[][] connections, int[] locations)
```

C#:

```
public string[] TotalDeliveryCost(int n, int[][] connections, int[] locations)
```

Python:

```
def total_delivery_cost(n, connections, locations):
```