

Problema D

No século de XXXV a humanidade colonizou o espaço graças ao domínio dos buracos de minhoca que lhe confere a possibilidade de viajar a grandes distâncias e a velocidades nunca vistas.

Explorando novos mundos, a humanidade estabeleceu colónias organizadas a volta dos buracos de minhoca em rede com a Terra. O centro desta estrutura de colónias é a Terra, e qualquer colónia tem um caminho único para a Terra, eventualmente passando por colónias intermédias, via os buracos de minhoca.

As colónias trocam matérias primas entre elas e com a Terra, via as rotas existentes. Contudo, viajar pelos buracos de minhoca tem o seu custo e cada colónia paga o preço forte com as transações que fazem com a Terra e com as restantes colónias para as quais conseguem viajar e, em consequências, transacionar.

A revolta aumenta e importa ao governo da Terra perceber que custo cada colónia realmente tem para suportar o seu comércio. Só assim consegue realmente agir a tempo, com justiça e apaziguar os ânimos.

Importa neste problema D calcular este custo por planeta.

O problema formula-se da seguinte forma: para cada colónia, qual é o maior custo das viagens que cada colónia conseguem organizar no contexto das suas relações comerciais?

Cada colónia consegue dois tipos de rotas:

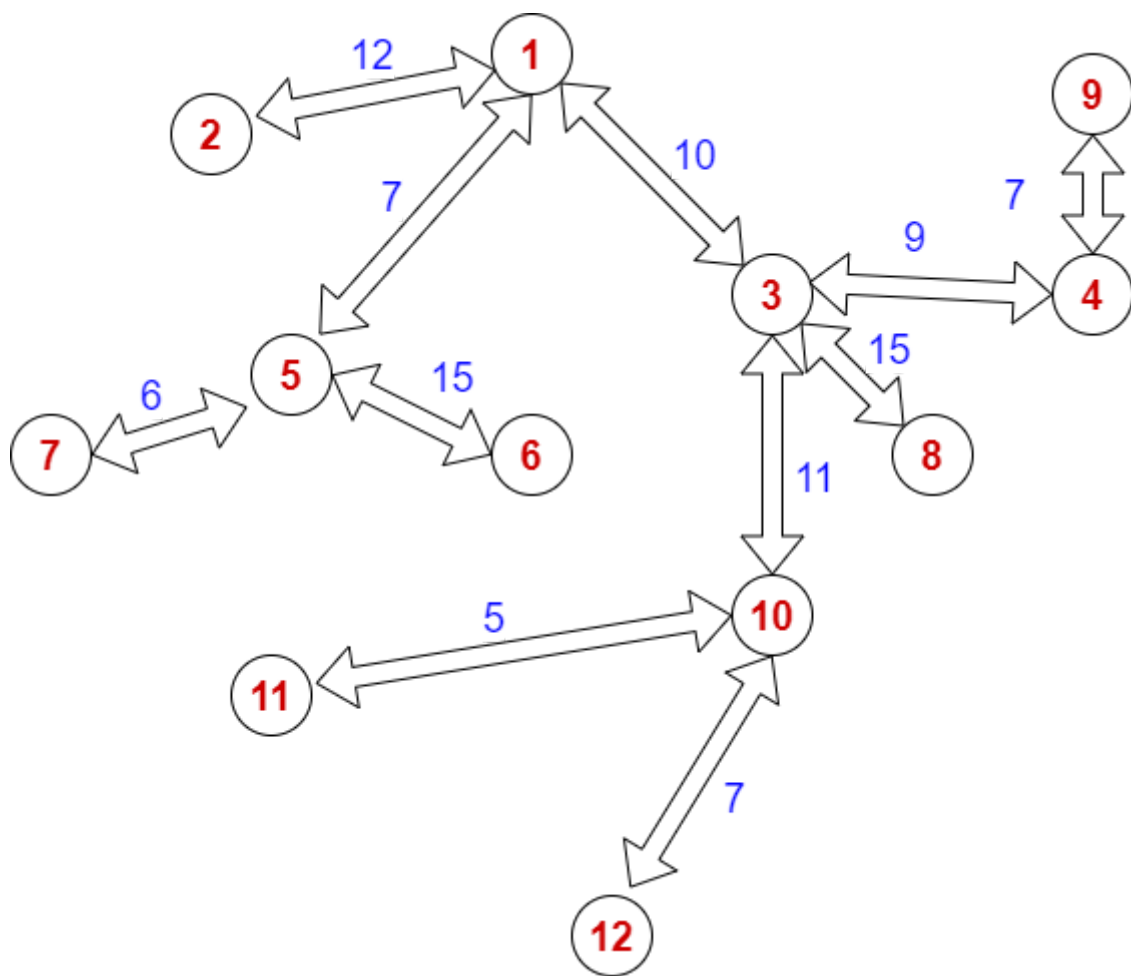
1. Viajar em direção à Terra e para qualquer planeta que se encontra a caminho da Terra.
2. Viajar para outras colónias que possuem um caminho para Terra que passa por ela.

Importa referir que a estrutura em rede aqui descrita forma um grafo acíclico não dirigido ponderado, ou seja uma árvore n-ária com pesos nos vértices, onde os nodos são as colónias, os vértices representam os buracos de minhoca, o peso dos vértices representa o custo da viagem pelo buraco de minhoca considerado e a raiz desta árvore é o planeta Terra.

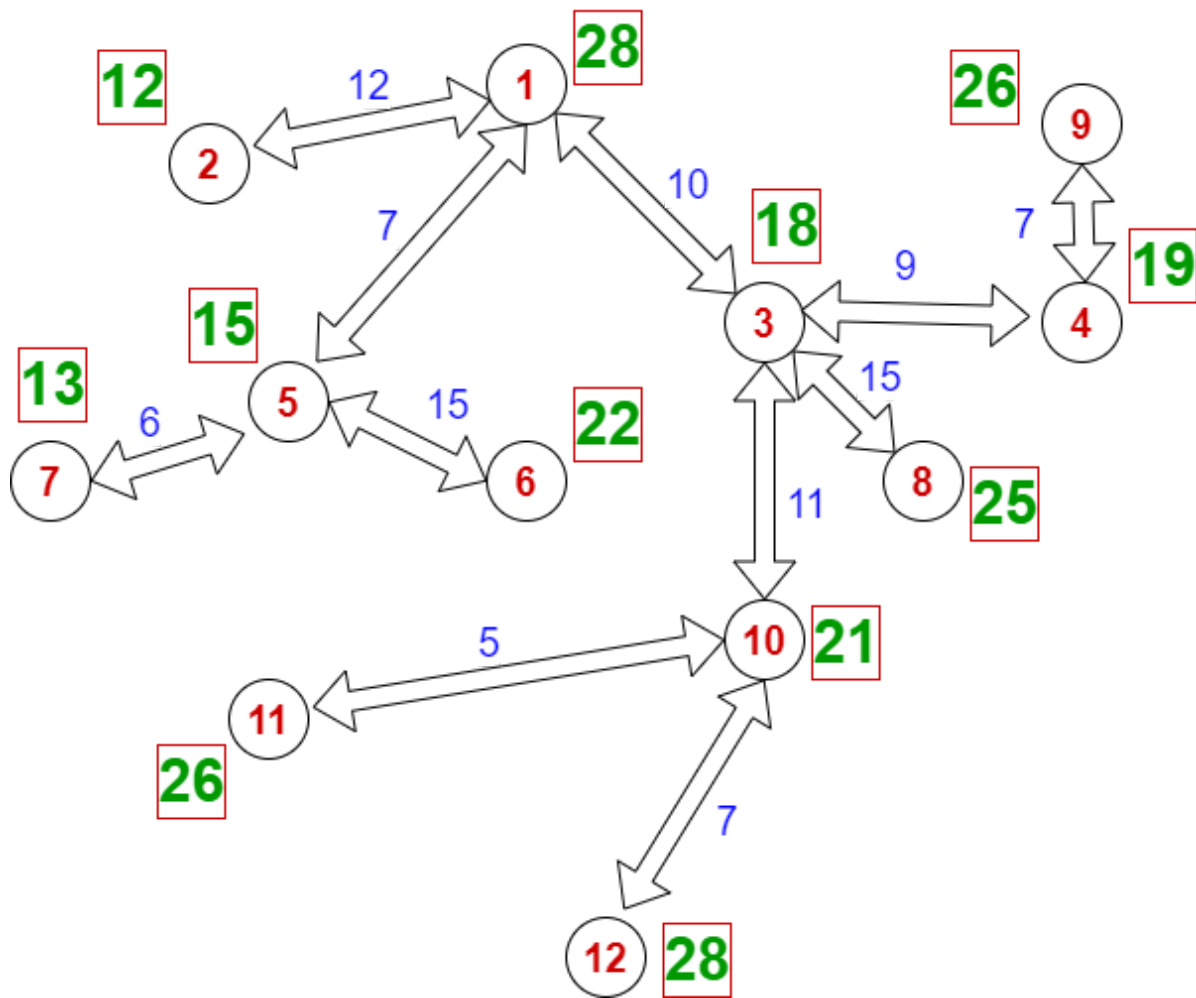
Assim uma colónia consegue uma rota para todas as colónias do caminho que parte da própria colónia até à Terra, e consegue uma rota para todas as colónias elementos da subárvore de que é raiz.

Parafraseando, pretendemos que seja calculados o custo máximo das rotas para todas as colónias da rede.

Por exemplo se o mapa de colónias, com a indicação dos buracos de minhoca e dos seus custos, que o planeta Terra, designada pelo nodo 1, coordena for o seguinte:



Então cada colônia sabe que o seu custo máximo é o seguinte:



Dicas:

1. Para representar comodamente a rede de colónias, é preciso definir o tipo das árvores n-árias em que cada nodo guarda igualmente a informação do seu pai.

Em OCaml e para esse efeito, várias soluções são possíveis. Uma delas é a seguinte:

```

1 type ntree = Node of int * int * (ntree*int) option * (ntree*int) list
2
3 (* se u v w são três colónias previamente definidas e ligadas à Terra que
4 sabemos ter um custo de 28, então podemos definir a Terra como: *)
5 let terra = Node (1,28,none,[ (u,12), (v,10), (w,7) ])
6
7 (* Terra: planeta numero 1, com peso 28, não tem pai, ligado aos planetas
8 u, v e w com buracos de minhoca de peso respectivo 12, 10 e 7 *)

```

Para ler os dados da rede, aconselha-se à leitura para um vetor ou uma tabela de hash que devidamente processado servirá de ponto de partida para construir a árvore.

Obviamente este tipo é dado de forma indicativa, é livre de escolher o tipo de dado que mais lhe convier.

2. Espera-se uma solução abaixo de quadrática, embora se tolere uma solução nestes termos.

Input

- Uma primeira linha com um inteiro indicando o numero de colônia. Se este número é n , então as colônias são numeradas de 1 até n e o planeta Terra é **sempre** o planeta 1.
- Uma segunda linha com o número m de buracos de minhoca. Há sempre buracos de minhoca suficientes para formar uma árvore tendo em conta as colônias.
- As m linhas seguintes introduzem os dados dos buracos de minhoca: três inteiros `a b c` em que `a` e `b` são dois inteiros identificando as colônias na extremidade do buraco de minhoca e `c` é o custo do referido buraco.

Output

A saída é constituída por n linhas com um inteiro. A linha i contém o custo da colônia i .

Exemplo de Input

```
1 12
2 11
3 1 2 12
4 1 3 10
5 1 5 7
6 3 4 9
7 3 8 15
8 3 10 11
9 4 9 7
10 5 6 15
11 5 7 6
12 10 11 5
13 10 12 7
14
```

Exemplo de Output

```
1 28
2 12
3 18
4 19
5 15
6 22
7 13
8 25
9 26
10 21
11 26
12 28
13
```

