

OJ 10099

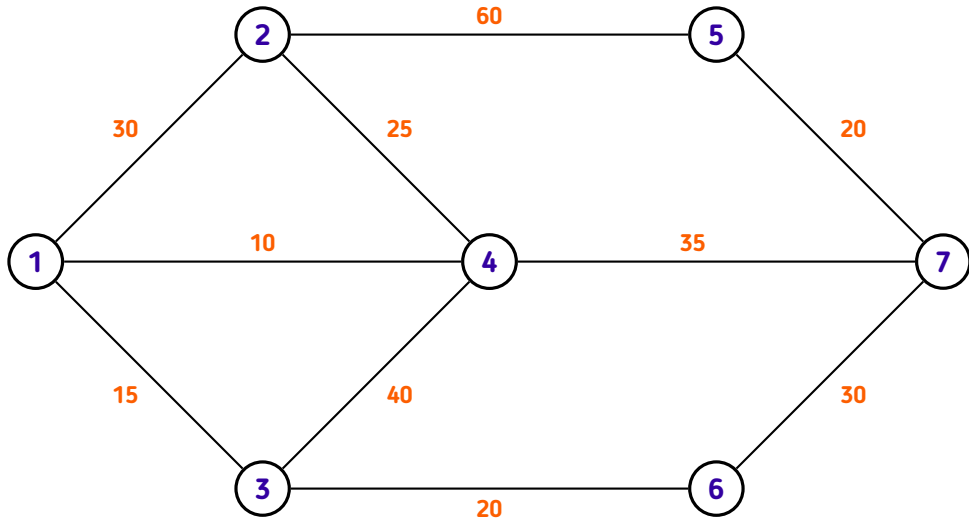
The Tourist Guide

Prof. Edson Alves

Faculdade UnB Gama

Mr. G. works as a tourist guide. His current assignment is to take some tourists from one city to another. Some two-way roads connect the cities. For each pair of neighboring cities there is a bus service that runs only between those two cities and uses the road that directly connects them. Each bus service has a limit on the maximum number of passengers it can carry. Mr. G. has a map showing the cities and the roads connecting them. He also has the information regarding each bus service. He understands that it may not always be possible for him to take all the tourists to the destination city in a single trip. For example, consider the following road map of 7 cities. The edges connecting the cities represent the roads and the number written on each edge indicates the passenger limit of the bus service that runs on that road.

O senhor G. trabalha como guia turístico. Sua tarefa atual é levar turistas de uma cidade a outra. As cidades são conectados por algumas estradas de mão dupla. Para cada par de cidades vizinhas há um serviço de ônibus que opera apenas entre estas duas cidades e que utiliza a estrada que as une diretamente. Cada serviço de ônibus tem um limite quanto ao número de passageiros que ele pode transportar. O senhor G. tem um mapa das cidades e das estradas que as conectam. Ele também tem informações a respeito dos serviços de ônibus. Ele entende que nem sempre é possível levar todos os turistas para a cidade de destino em uma única viagem. Por exemplo, considere o seguinte mapa, com 7 cidades. As arestas conectando as cidades representam as estradas e o número associado a cada aresta indica o limite de passageiros do serviço de ônibus que opera naquela estrada.



Now, if he wants to take 99 tourists from city 1 to city 7, he will require at least 5 trips and the route he should take is : 1 - 2 - 4 - 7.

But, Mr. G. finds it difficult to find the best route all by himself so that he may be able to take all the tourists to the destination city in minimum number of trips. So, he seeks your help.

Se ele quer levar 99 turistas da cidade 1 para a cidade 7, ele precisará no mínimo de 5 viagens e a rota que ele deverá seguir é: 1 - 2 - 4 - 7.

Mas o senhor G. descobriu que é difícil determinar a melhor rota por conta própria, de modo que ele possa levar todos os turistas ao destino deles no menor número de viagens possível. Assim, ele pediu a sua ajuda.

Input

The input will contain one or more test cases. The first line of each test case will contain two integers: N ($N \leq 100$) and R representing respectively the number of cities and the number of road segments. Then R lines will follow each containing three integers: C_1, C_2 and P . C_1 and C_2 are the city numbers and P ($P > 1$) is the limit on the maximum number of passengers to be carried by the bus service between the two cities. City numbers are positive integers ranging from 1 to N . The $(R + 1)$ -th line will contain three integers: S, D and T representing respectively the starting city, the destination city and the number of tourists to be guided.

The input will end with two zeroes for N and R .

Entrada

A entrada é composta por um ou mais casos de teste. A primeira linha de cada caso de teste contém dois inteiros: N ($N \leq 100$) e R , representando o número de cidades e o número de estradas, respectivamente. As R linhas seguintes contém três inteiros: C_1, C_2 e P . C_1 e C_2 são os números das cidades e P ($P > 1$) é o número máximo de passageiros que podem ser transportados pelo serviço de ônibus entre as duas cidades. Os números das cidades são inteiros positivos entre 1 e N . A $(R+1)$ -ésima linha contém três inteiros: S, D e T , representando o ponto de partida, o destino e o número de turistas a serem guiados, respectivamente.

A entrada termina com dois zeros para N e R .

Output

For each test case in the input first output the scenario number. Then output the minimum number of trips required for this case on a separate line. Print a blank line after the output of each test case.

Saída

Para cada caso de teste da entrada imprima o número do cenário. Então imprima o número mínimo de viagens necessárias neste caso em uma linha. Imprima uma linha em branco após a saída de cada caso de teste.

Exemplo de entrada e saída

Exemplo de entrada e saída

7 10

Exemplo de entrada e saída

7 10



de cidades

Exemplo de entrada e saída

7 10
↑
de estradas

Exemplo de entrada e saída

7 10

1

2

5

4

7

3

6

Exemplo de entrada e saída

7 10

1 2 30

1

2

5

4

7

3

6

Exemplo de entrada e saída

7 10
1 2 30
↑
cidade C_1

1

2

5

4

7

3

6

Exemplo de entrada e saída

7 10
1 2 30
↑
cidade C_2

1

2

5

4

7

3

6

Exemplo de entrada e saída

7 10
1 2 30
↑
máximo de turistas

1

2

5

4

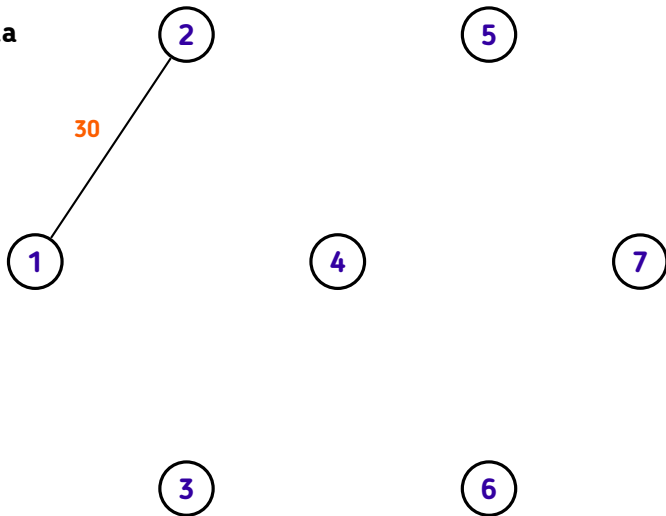
7

3

6

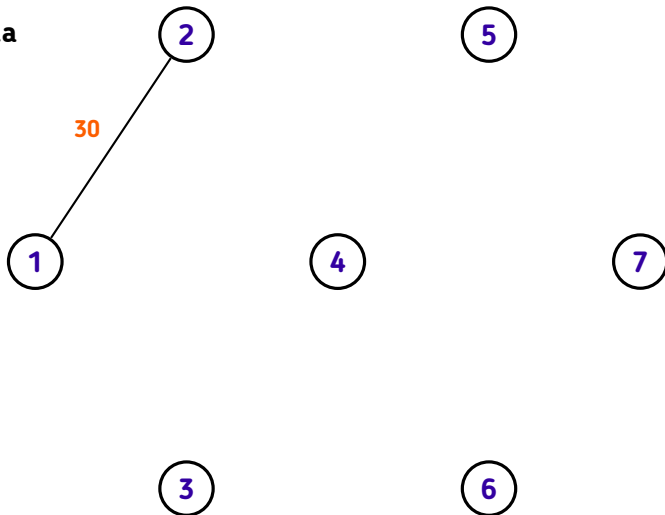
Exemplo de entrada e saída

7 10
1 2 30



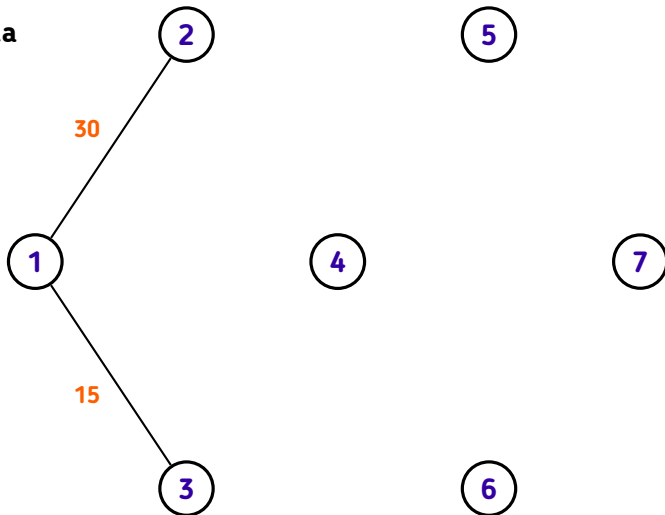
Exemplo de entrada e saída

7 10
1 2 30
1 3 15



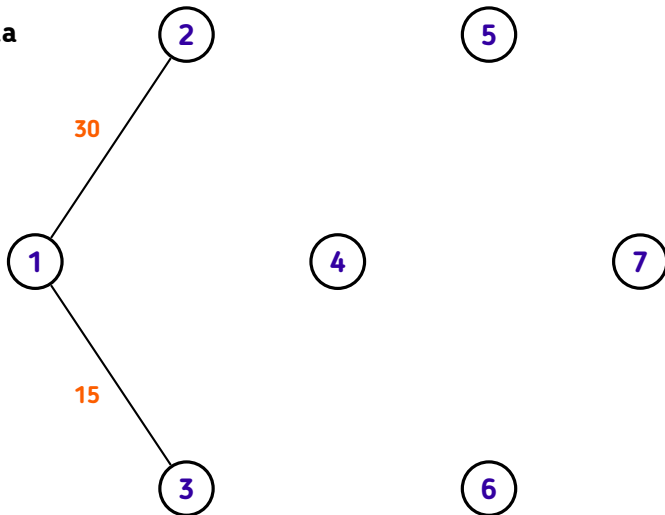
Exemplo de entrada e saída

7 10
1 2 30
1 3 15



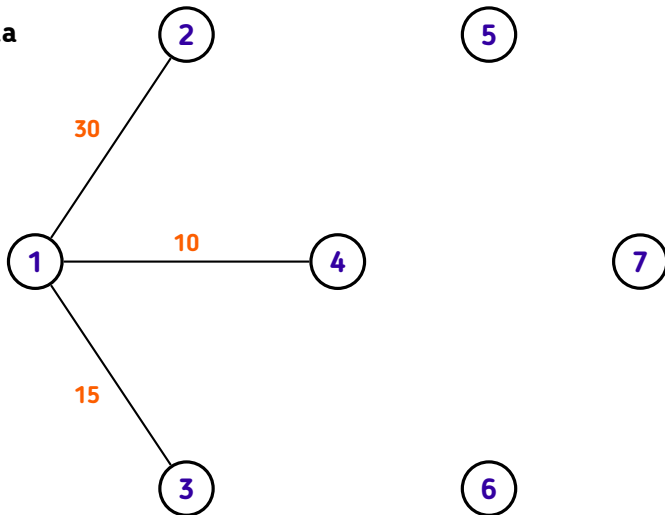
Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10



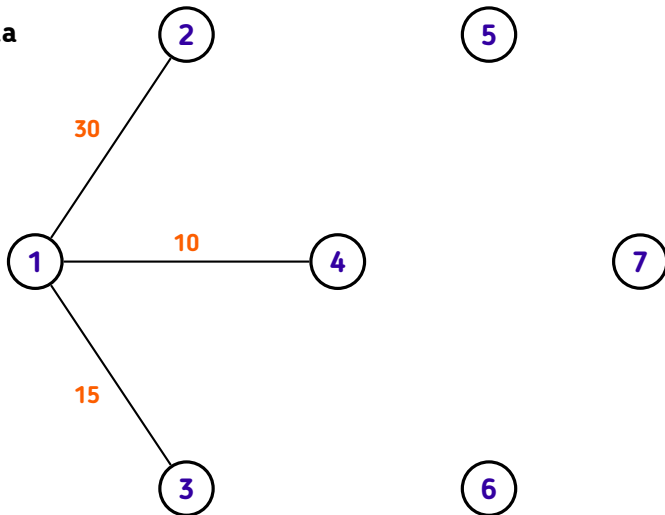
Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10



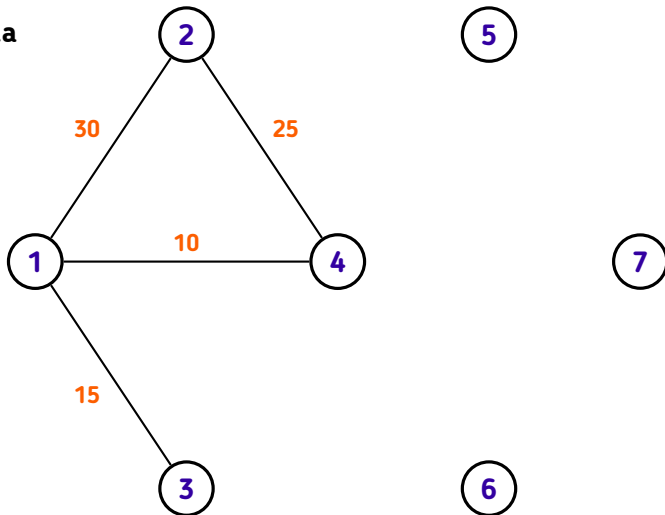
Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10
2 4 25



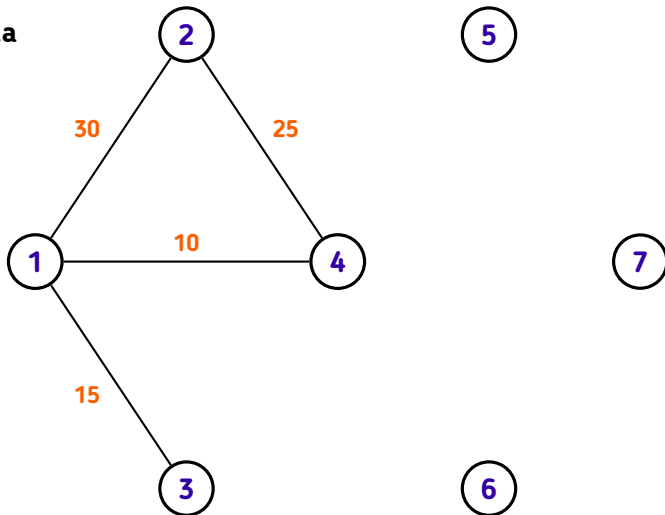
Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10
2 4 25



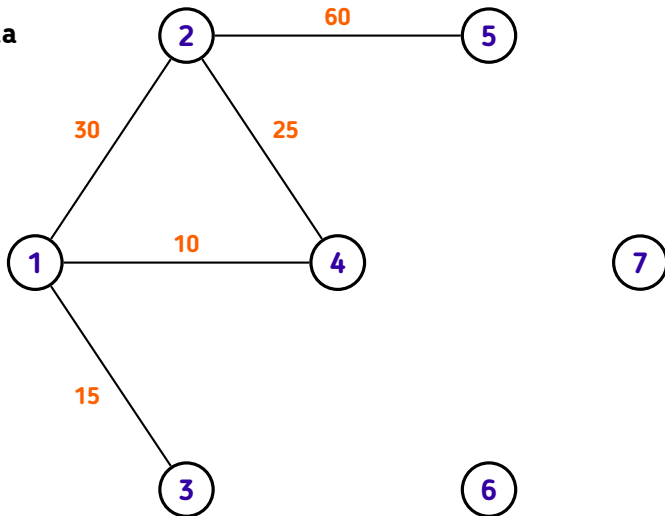
Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10
2 4 25
2 5 60



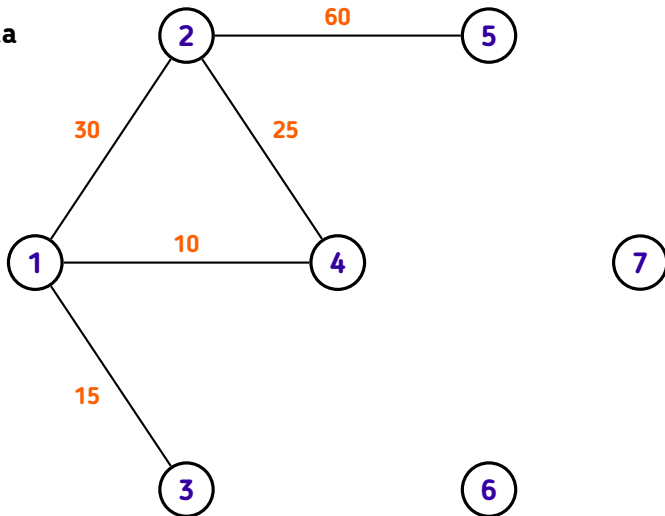
Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10
2 4 25
2 5 60
2 5 60



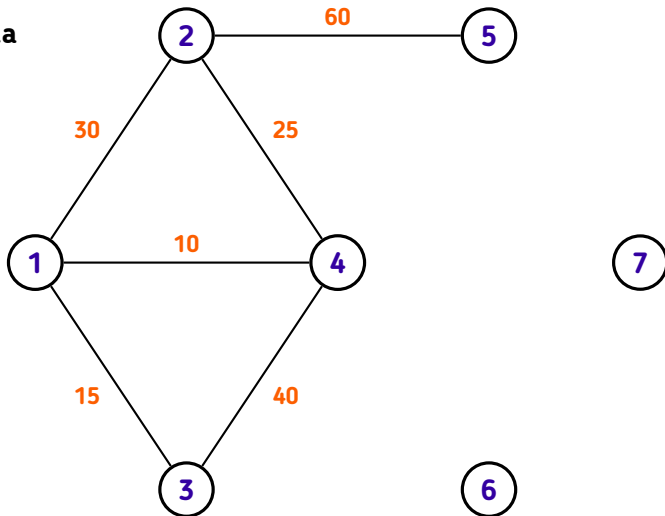
Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10
2 4 25
2 5 60
3 4 40



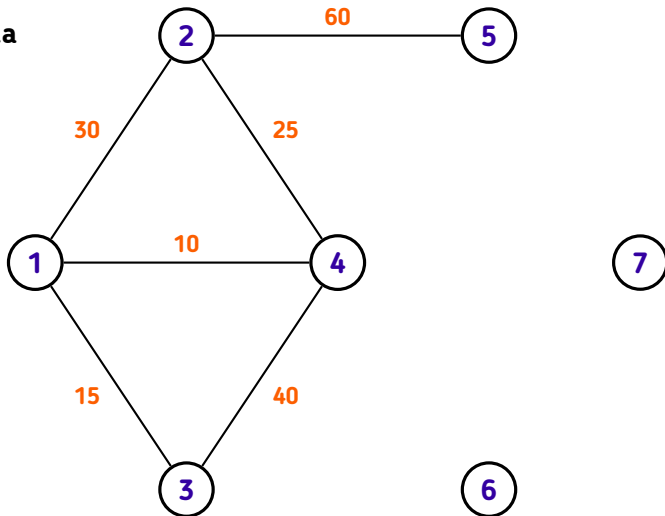
Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10
2 4 25
2 5 60
3 4 40



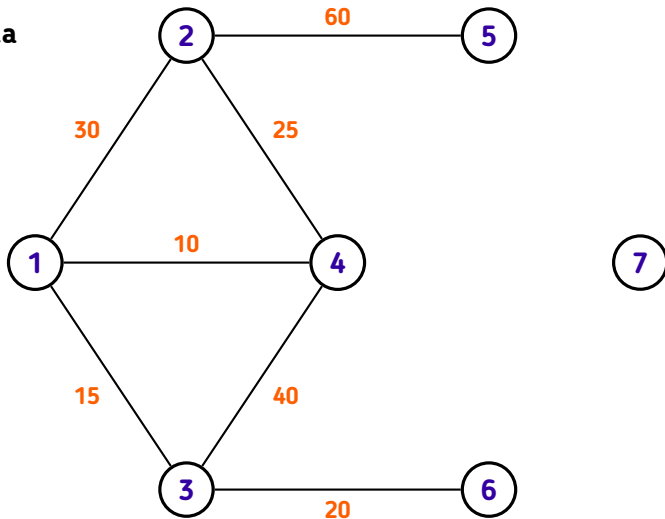
Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10
2 4 25
2 5 60
3 4 40
3 6 20



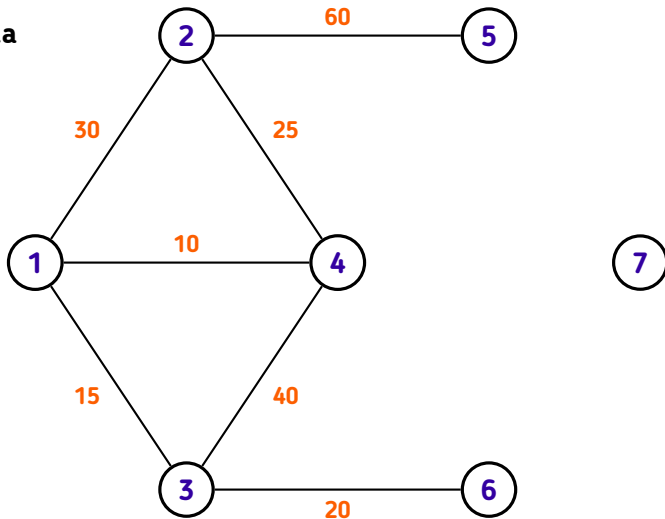
Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10
2 4 25
2 5 60
3 4 40
3 6 20



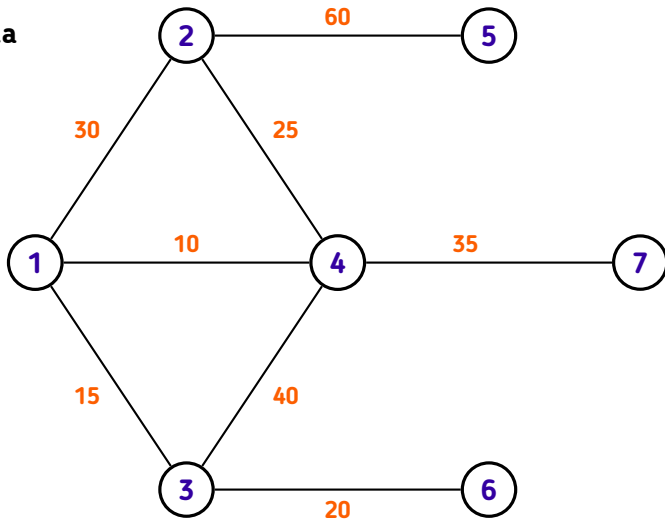
Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10
2 4 25
2 5 60
3 4 40
3 6 20
4 7 35



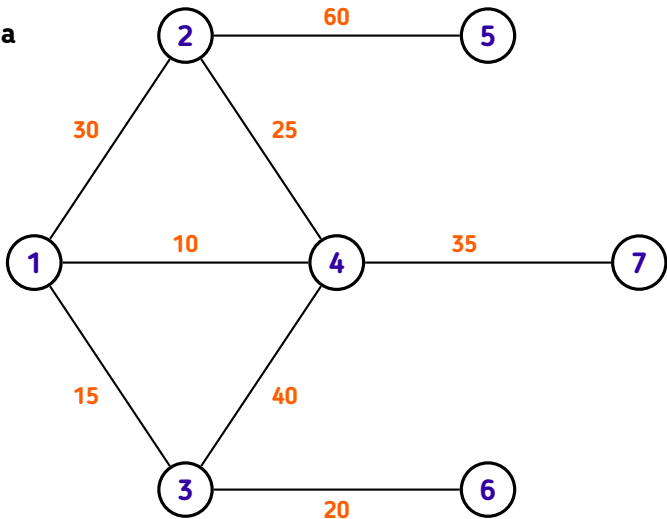
Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10
2 4 25
2 5 60
3 4 40
3 6 20
4 7 35



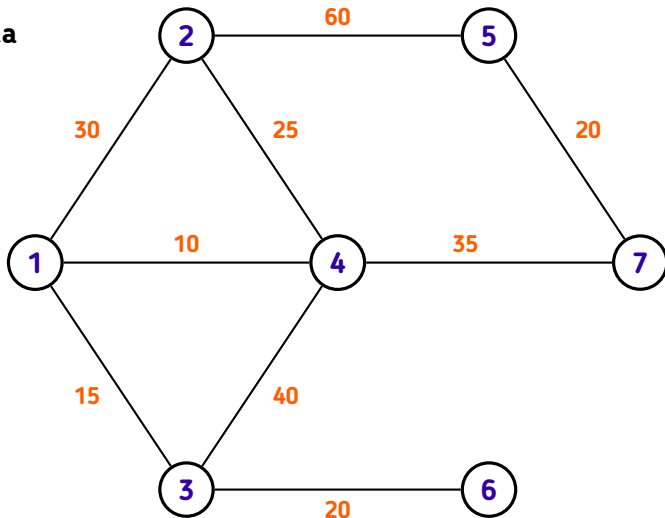
Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10
2 4 25
2 5 60
3 4 40
3 6 20
4 7 35
5 7 20



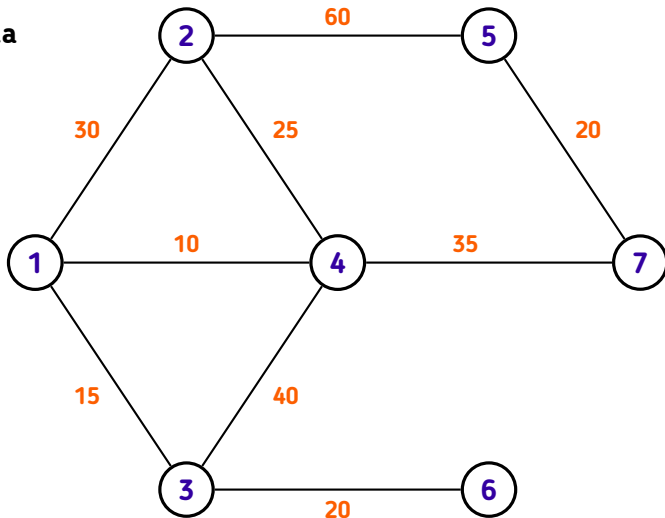
Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10
2 4 25
2 5 60
3 4 40
3 6 20
4 7 35
5 7 20



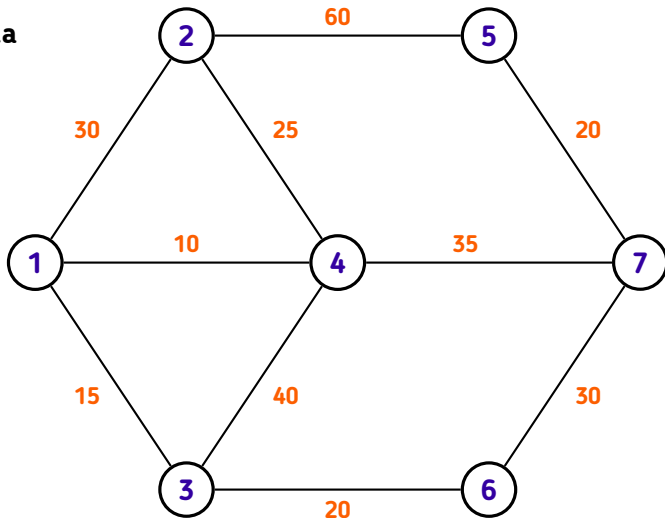
Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10
2 4 25
2 5 60
3 4 40
3 6 20
4 7 35
5 7 20
6 7 30



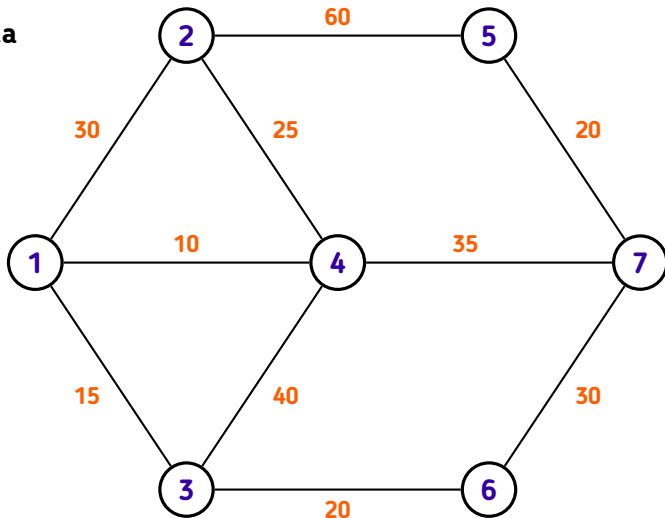
Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10
2 4 25
2 5 60
3 4 40
3 6 20
4 7 35
5 7 20
6 7 30



Exemplo de entrada e saída

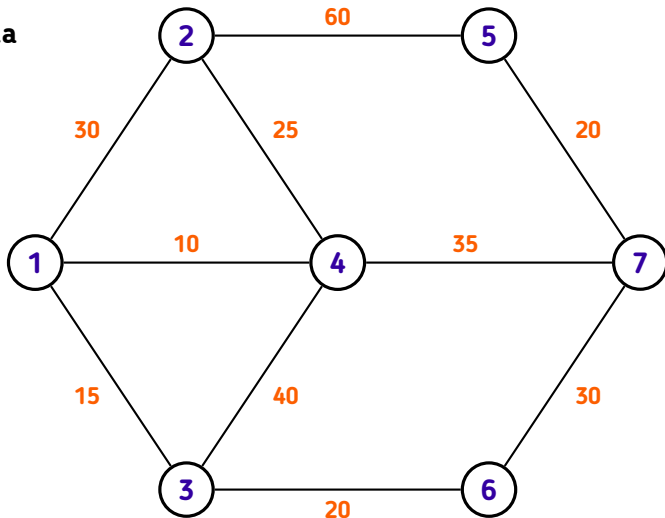
7 10
1 2 30
1 3 15
1 4 10
2 4 25
2 5 60
3 4 40
3 6 20
4 7 35
5 7 20
6 7 30
1 7 99



Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10
2 4 25
2 5 60
3 4 40
3 6 20
4 7 35
5 7 20
6 7 30
1 7 99

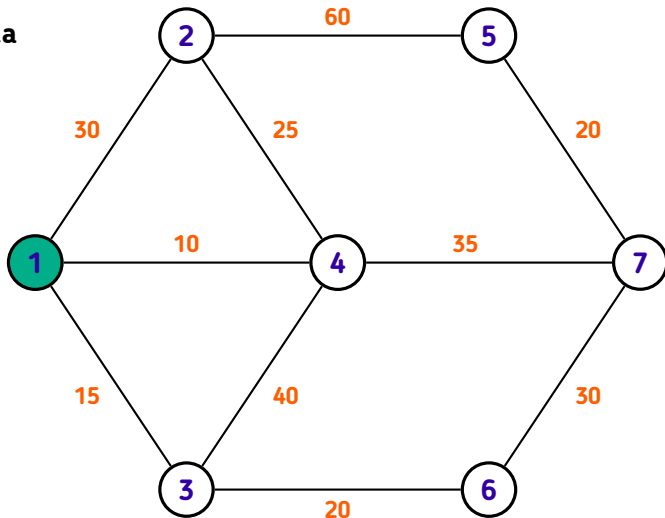
└─> ponto de partida



Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10
2 4 25
2 5 60
3 4 40
3 6 20
4 7 35
5 7 20
6 7 30
1 7 99

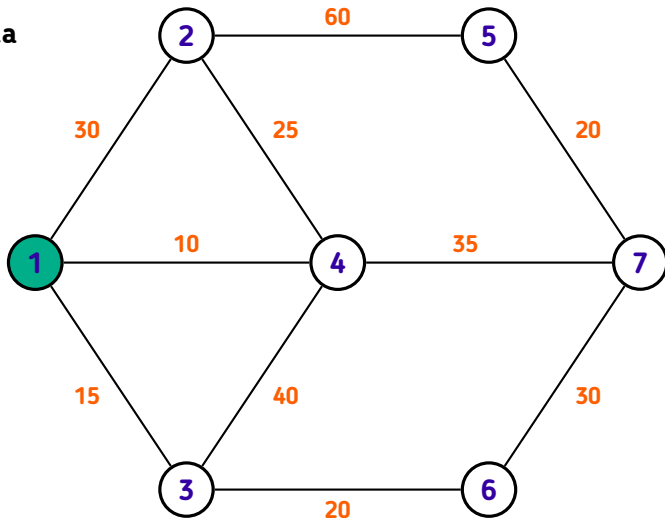
└─> ponto de partida



Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10
2 4 25
2 5 60
3 4 40
3 6 20
4 7 35
5 7 20
6 7 30
1 7 99

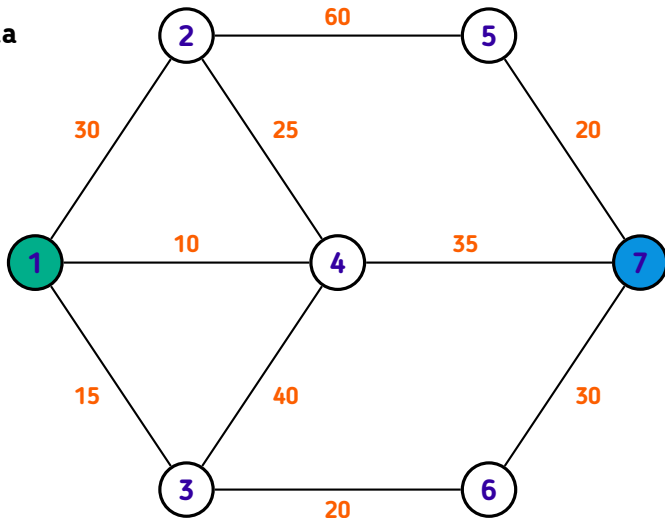
└───> destino



Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10
2 4 25
2 5 60
3 4 40
3 6 20
4 7 35
5 7 20
6 7 30
1 7 99

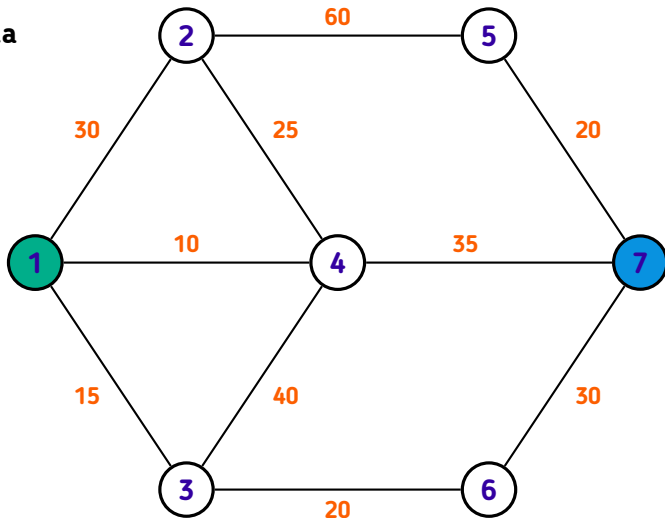
└───> destino



Exemplo de entrada e saída

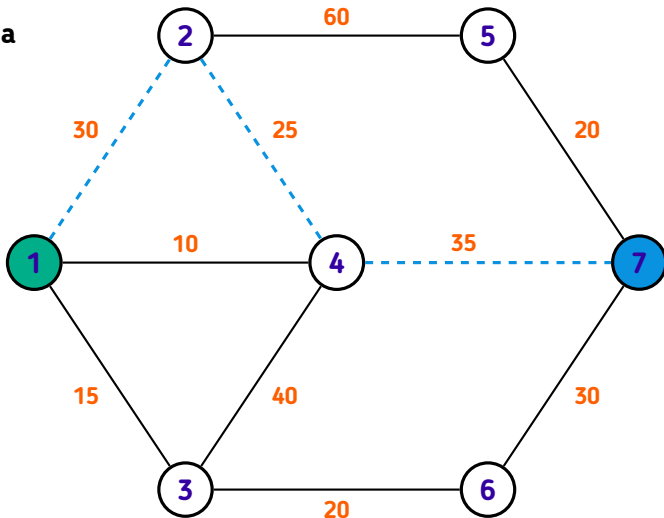
7 10
1 2 30
1 3 15
1 4 10
2 4 25
2 5 60
3 4 40
3 6 20
4 7 35
5 7 20
6 7 30
1 7 99

└─ # de turistas



Exemplo de entrada e saída

7 10
1 2 30
1 3 15
1 4 10
2 4 25
2 5 60
3 4 40
3 6 20
4 7 35
5 7 20
6 7 30
1 7 99



Exemplo de entrada e saída

7 10

1 2 30

1 3 15

1 4 10

2 4 25

2 5 60

3 4 40

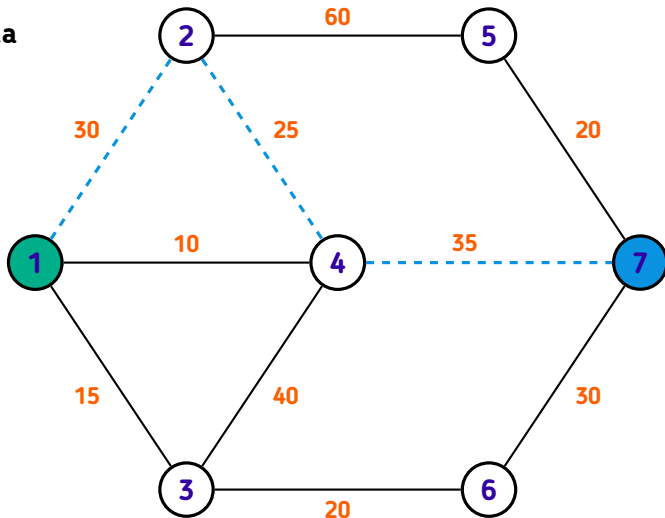
3 6 20 → 5

4 7 35

5 7 20

6 7 30

1 7 99



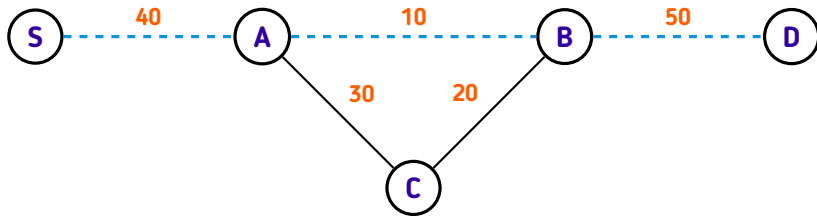
Solução

Solução

Considere um caminho entre a partida S e o destino D

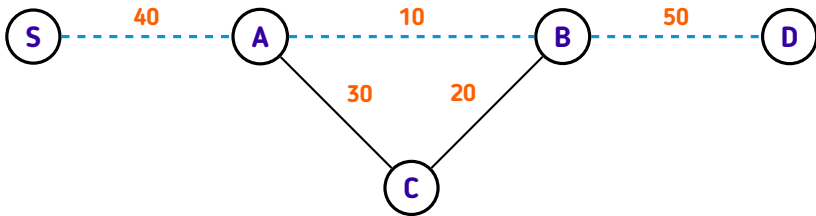
Solução

Considere um caminho entre a partida S e o destino D



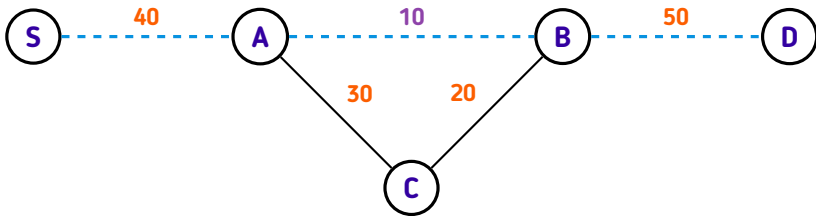
Solução

O máximo de passageiros é determinado pela aresta de menor peso



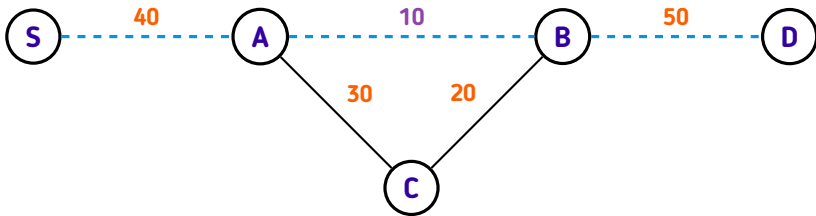
Solução

O máximo de passageiros é determinado pela aresta de menor peso



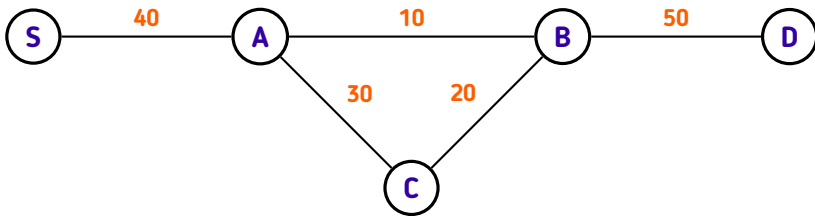
Solução

O máximo é dado por este peso mínimo menos um, por conta do guia, que ocupa uma posição no ônibus



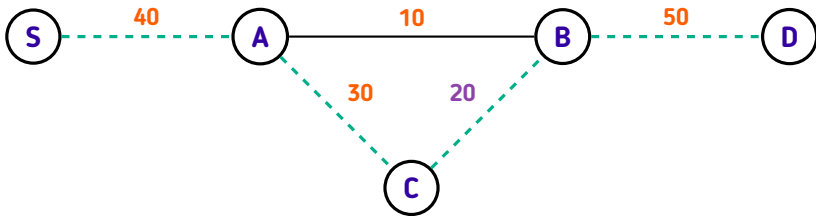
Solução

O problema consiste em determinar o caminho de S a D que maximiza este peso mínimo, ou seja, um problema *maximin*



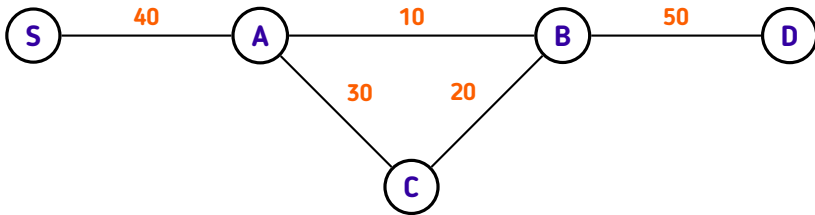
Solução

O problema consiste em determinar o caminho de S a D que maximiza este peso mínimo, ou seja, um problema *maximin*



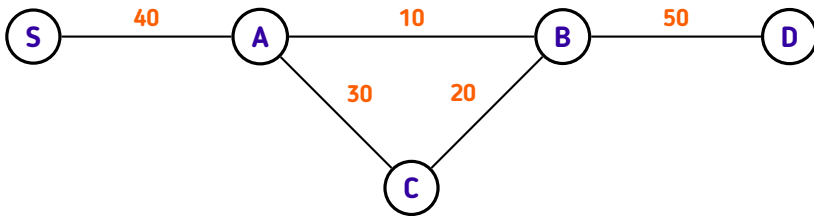
Solução

A MST resolve o problema *maximin*.



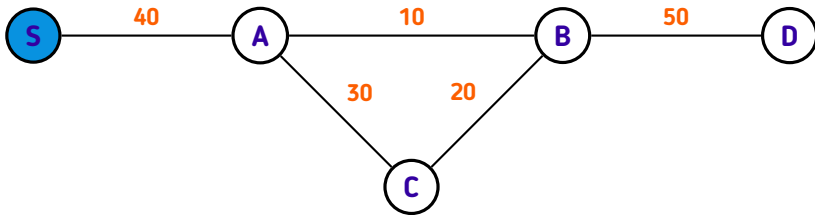
Solução

O algoritmo de Prim pode ser modificado para computar $\text{maximin}(S, u)$ para todos os vértices do componente conectado que contém S



Solução

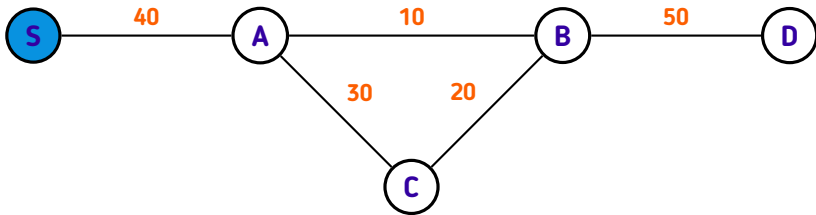
Seja u o próximo vértice a ser adicionado a C



Solução

Se a aresta que conecta u a C é (u, S, w) , então

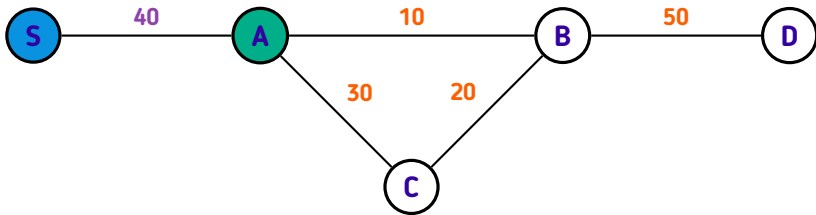
$$\text{maximin}(\mathbf{S}, u) = w$$



Solução

Se a aresta que conecta u a C é (u, S, w) , então

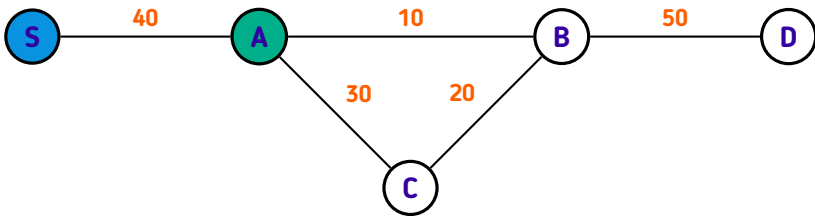
$$\text{maximin}(\mathbf{S}, u) = w$$



Solução

Se a aresta que conecta u a C é (u, v, w) , então

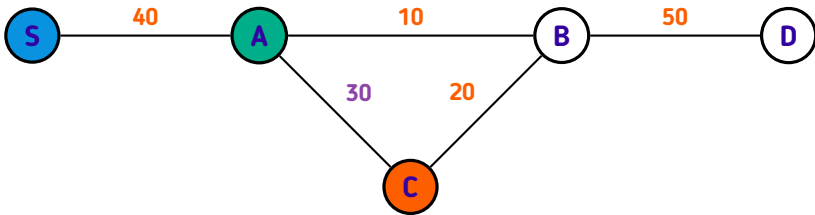
$$\text{maximin}(S, u) = \min \{ \text{maximin}(S, v), w \}$$



Solução

Se a aresta que conecta u a C é (u, v, w) , então

$$\text{maximin}(S, u) = \min \{ \text{maximin}(S, v), w \}$$



```
vector<int> maximin(int s, int N)
{
    set<int> C;
    C.insert(s);

    priority_queue<edge> pq;
    vector<int> ms(N + 1, oo);

    for (auto [v, w] : adj[s])
        pq.push(edge(w, s, v));

    while ((int) C.size() < N)
    {
        auto [w, u, v] = pq.top();
        pq.pop();

        if (C.count(v))
            continue;

        C.insert(v);
    }
}
```

```
ms[v] = u == s ? w : min(w, ms[u]);
```

```
for (auto [r, c] : adj[v])
```

```
    pq.push(edge(c, v, r));
```

```
}
```

```
return ms;
```

```
}
```

```
int solve(int N, int S, int D, int T)
```

```
{
```

```
    auto ms = maximin(S, N);
```

```
    auto M = ms[D];
```

```
    return (T + M - 2)/(M - 1);
```

```
}
```