

Educational Codeforces Round 33

Problem C – Rumor

Prof. Edson Alves

Faculdade UnB Gama

Vova promised himself that he would never play computer games... But recently Firestorm – a well-known game developing company – published their newest game, World of Warcraft, and it became really popular. Of course, Vova started playing it.

Now he tries to solve a quest. The task is to come to a settlement named Overcity and spread a rumor in it.

Vova knows that there are n characters in Overcity. Some characters are friends to each other, and they share information they got. Also Vova knows that he can bribe each character so he or she starts spreading the rumor; i -th character wants c_i gold in exchange for spreading the rumor. When a character hears the rumor, he tells it to all his friends, and they start spreading the rumor to their friends (for free), and so on.

Vova prometeu a si mesmo que ele nunca jogaria jogos de computador... Mas recentemente Firestorm – uma companhia de desenvolvimento de jogos bem conhecida – publicou seu jogo mais recente, World of Warcraft, e ele se tornou realmente popular. Evidentemente, Vova começou a jogá-lo.

Agora ele está tentando finalizar uma missão. A tarefa é ir a um assentamento chamado Overcity e espalhar um rumor nele.

Vova sabe que há n personagens em Overcity. Algumas personagens são amigas umas das outras, e elas compartilham as informações que obtém. Vova também sabe que ele pode subornar uma personagem de modo que ele ou ela inicie o rumor; a i -ésima personagem quer c_i moedas de ouro para espalhar o rumor. Quando uma personagem ouve um rumor, ela o espalha a todos seus amigos, e estes repassam o rumor a seus amigos (de graça), e assim por diante.

The quest is finished when all n characters know the rumor. What is the minimum amount of gold Vova needs to spend in order to finish the quest?

Take a look at the notes if you think you haven't understood the problem completely.

A missão termina quando todas n personagens tem conhecimento do rumor. Qual é a menor quantidade de moedas de ouro que Vova precisa gastar para finalizar a missão?

Observe as notas se você não tenha entendido o problema completamente.

Input

The first line contains two integer numbers n and m ($1 \leq n \leq 10^5, 0 \leq m \leq 10^5$) – the number of characters in Overcity and the number of pairs of friends.

The second line contains n integer numbers c_i ($0 \leq c_i \leq 10^9$) – the amount of gold i -th character asks to start spreading the rumor.

Then m lines follow, each containing a pair of numbers (x_i, y_i) which represent that characters x_i and y_i are friends ($1 \leq x_i, y_i \leq n, x_i \neq y_i$). It is guaranteed that each pair is listed at most once.

Output

Print one number – the minimum amount of gold Vova has to spend in order to finish the quest.

Entrada

A primeira linha contém dois inteiros n e m ($1 \leq n \leq 10^5, 0 \leq m \leq 10^5$) – o número de personagens em Overcity e o número de pares de amigos.

A segunda linha contém n números inteiros c_i ($0 \leq c_i \leq 10^9$) – a quantidade de moedas de ouro que a i -ésima personagem quer para começar a espalhar o rumor.

As m linhas seguintes contém, cada uma, um par de números (x_i, y_i) que indicam que as personagens x_i e y_i são amigas ($1 \leq x_i, y_i \leq n, x_i \neq y_i$). É garantido que cada par é listado, no máximo, uma vez.

Output

Imprima um número – a quantidade mínima de moedas de ouro que Vova tem que gastar para completar a missão.

Exemplo de entrada e saída

Exemplo de entrada e saída

5 2

Exemplo de entrada e saída

5 2



de personagens

Exemplo de entrada e saída

5 2 \longrightarrow *# de pares de amigos*
 \uparrow
de personagens

Exemplo de entrada e saída

5 2

2 5 3 4 8

Exemplo de entrada e saída

5 2

2 5 3 4 8 \longrightarrow *custo individual para iniciar o rumor*

Exemplo de entrada e saída

5 2

2 5 3 4 8

1^2

2^5

3^3

4^4

5^8

Exemplo de entrada e saída

5 2

2 5 3 4 8

1 4

1^2

2^5

3^3

4^4

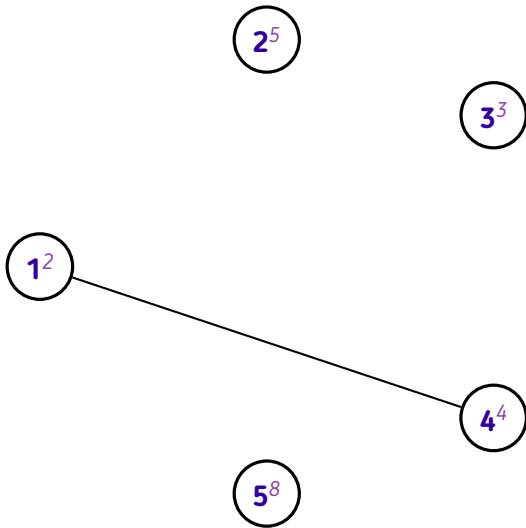
5^8

Exemplo de entrada e saída

5 2

2 5 3 4 8

1 4



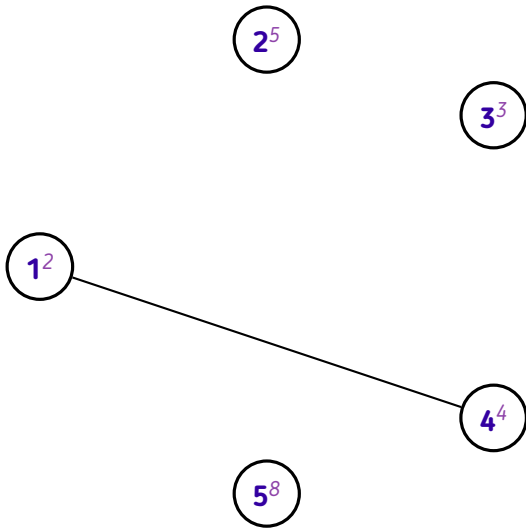
Exemplo de entrada e saída

5 2

2 5 3 4 8

1 4

4 5



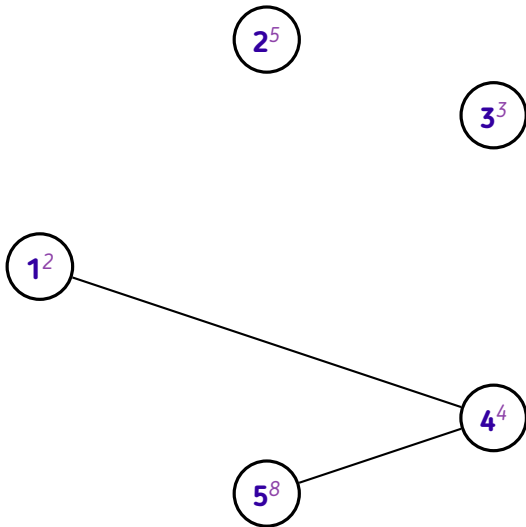
Exemplo de entrada e saída

5 2

2 5 3 4 8

1 4

4 5



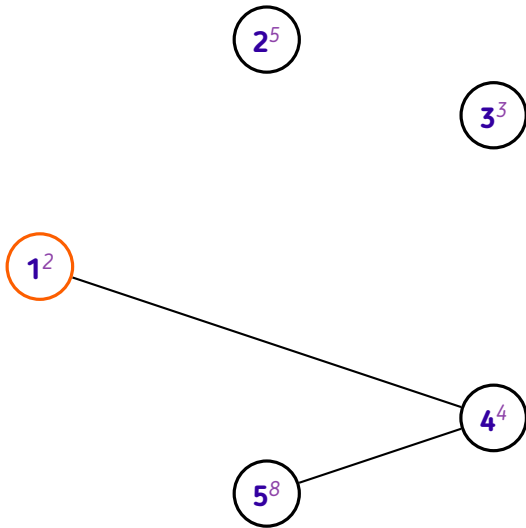
Exemplo de entrada e saída

5 2

2 5 3 4 8

1 4

4 5



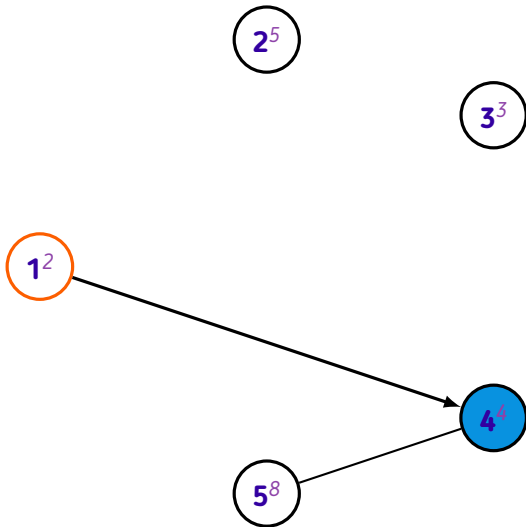
Exemplo de entrada e saída

5 2

2 5 3 4 8

1 4

4 5



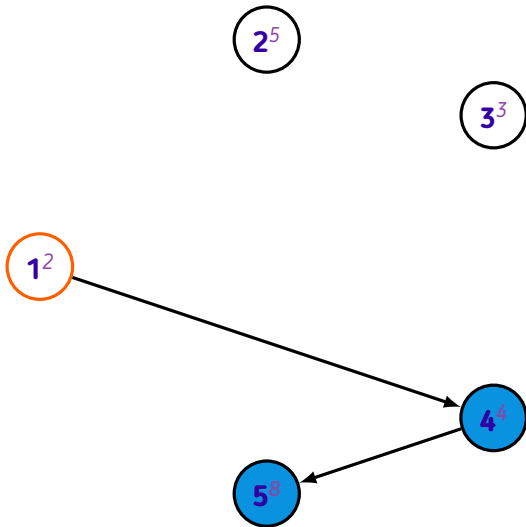
Exemplo de entrada e saída

5 2

2 5 3 4 8

1 4

4 5



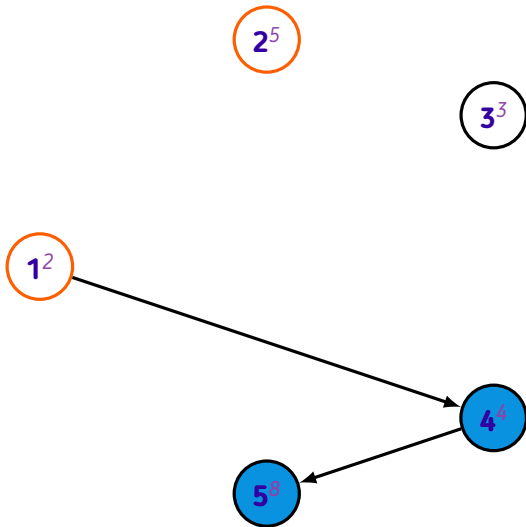
Exemplo de entrada e saída

5 2

2 5 3 4 8

1 4

4 5



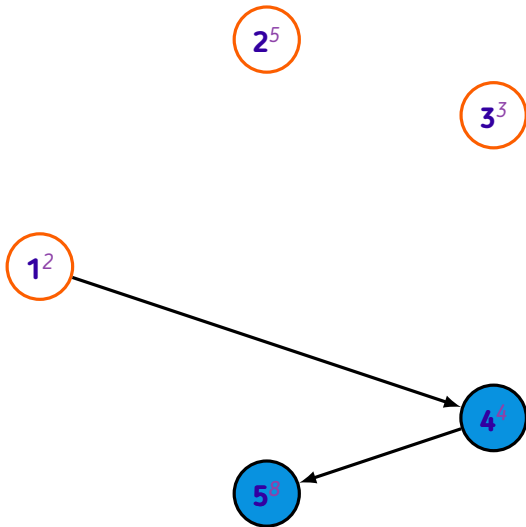
Exemplo de entrada e saída

5 2

2 5 3 4 8

1 4

4 5



Exemplo de entrada e saída

5 2

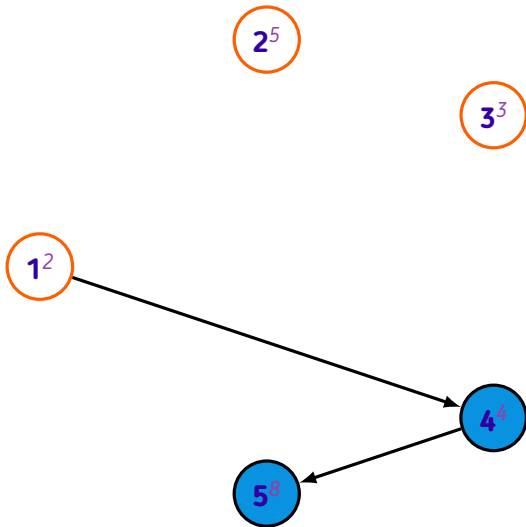
2 5 3 4 8

1 4

4 5



$$2 + 5 + 3 = 10$$



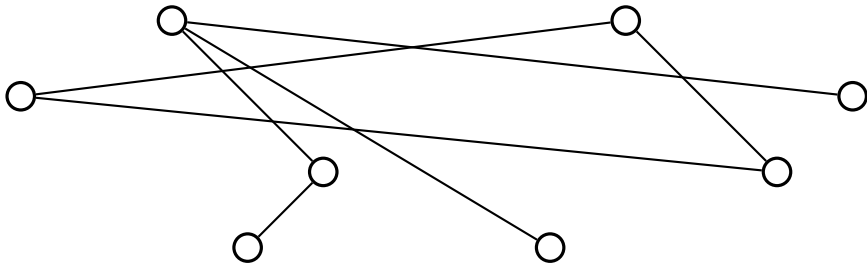
Solução

Solução

Cada vértice u pago espalha o rumor para todos os vértices que estão no componente conectado que contém u .

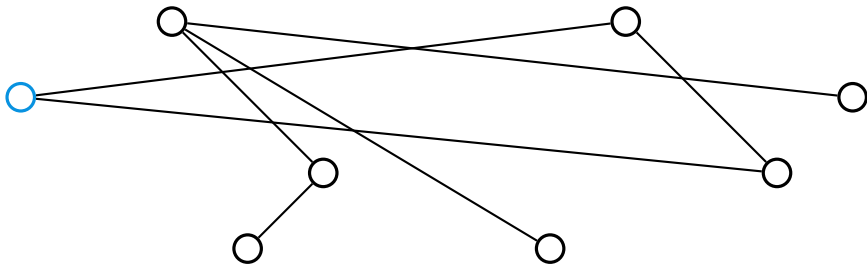
Solução

Cada vértice u pago espalha o rumor para todos os vértices que estão no componente conectado que contém u .



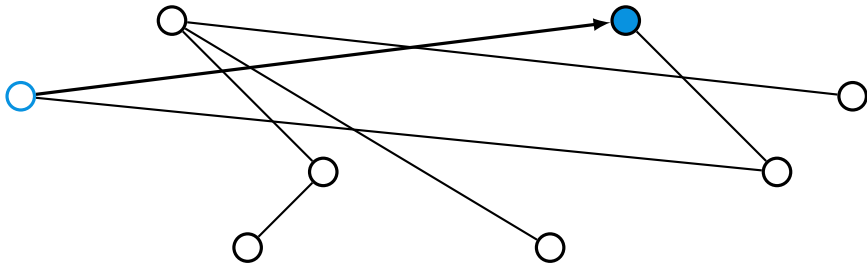
Solução

Cada vértice u pago espalha o rumor para todos os vértices que estão no componente conectado que contém u .



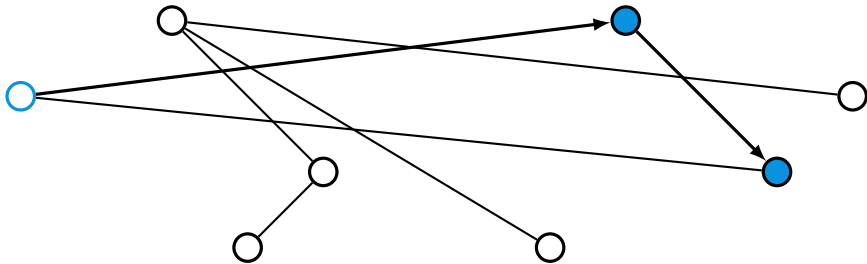
Solução

Cada vértice u pago espalha o rumor para todos os vértices que estão no componente conectado que contém u .



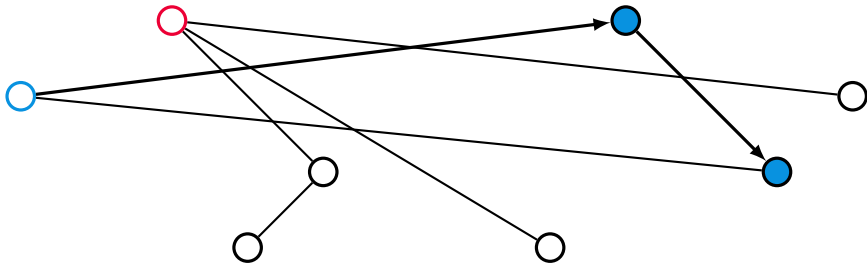
Solução

Cada vértice u pago espalha o rumor para todos os vértices que estão no componente conectado que contém u .



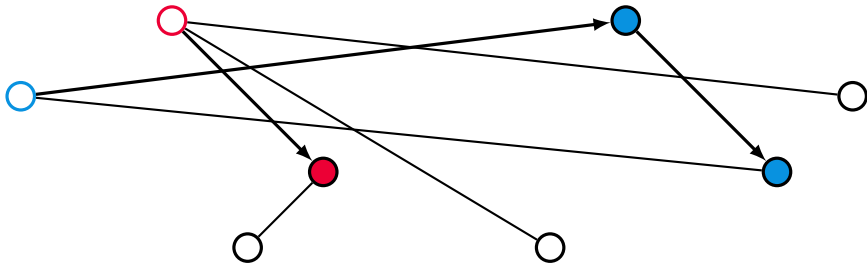
Solução

Cada vértice u pago espalha o rumor para todos os vértices que estão no componente conectado que contém u .



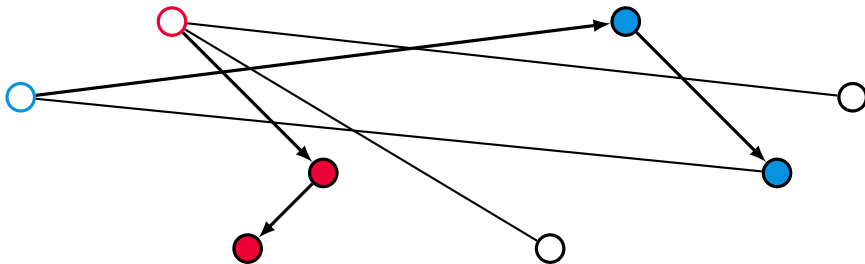
Solução

Cada vértice u pago espalha o rumor para todos os vértices que estão no componente conectado que contém u .



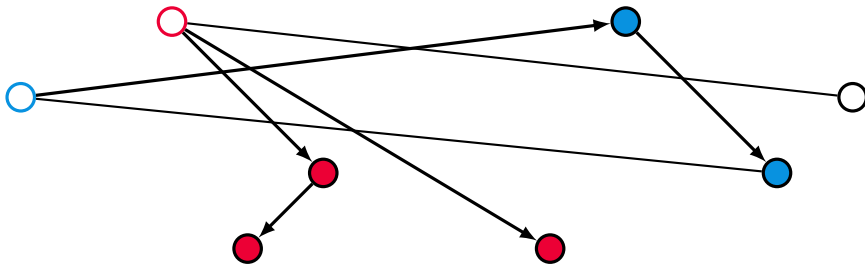
Solução

Cada vértice u pago espalha o rumor para todos os vértices que estão no componente conectado que contém u .



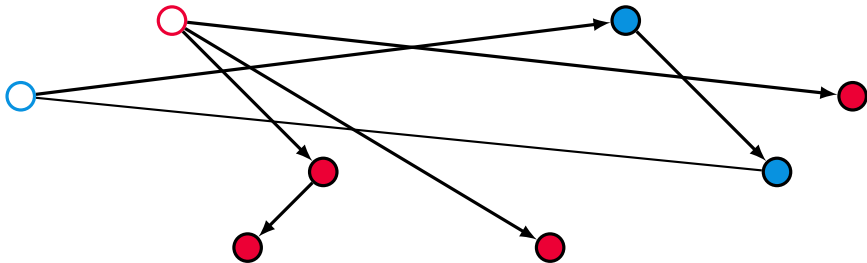
Solução

Cada vértice u pago espalha o rumor para todos os vértices que estão no componente conectado que contém u .



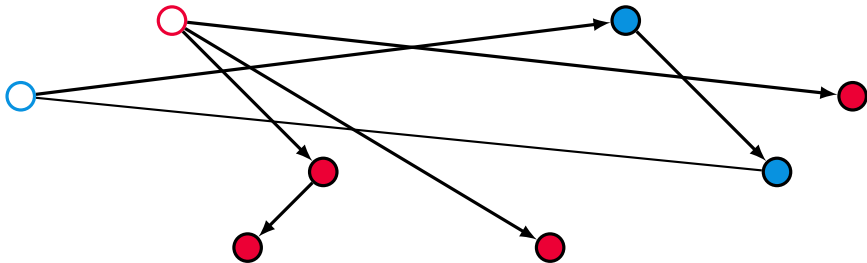
Solução

Cada vértice u pago espalha o rumor para todos os vértices que estão no componente conectado que contém u .



Solução

Logo, basta pagar o vértice de menor custo em cada componente conectado.



```
void dfs(int u, const function<void(int)>& process)
{
    if (visited[u])
        return;

    visited[u] = true;
    process(u);

    for (auto v : adj[u])
        dfs(v, process);
}
```

```
ll solve(int N, const vector<ll>& cs)
{
    ll ans = 0;

    for (int u = 1; u <= N; ++u)
    {
        if (visited[u])
            continue;

        ll cost = oo;

        dfs(u, [&](int u) { cost = min(cost, cs[u]); });

        ans += cost;
    }

    return ans;
}
```