

Codeforces Round #250 (Div. 2)

Problem C – The Child and Toy

Prof. Edson Alves

Faculdade UnB Gama

On Children's Day, the child got a toy from Delayyy as a present. However, the child is so naughty that he can't wait to destroy the toy.

The toy consists of n parts and m ropes. Each rope links two parts, but every pair of parts is linked by at most one rope. To split the toy, the child must remove all its parts. The child can remove a single part at a time, and each remove consume an energy. Let's define an energy value of part i as v_i . The child spend $v_{f_1} + v_{f_2} + \dots + v_{f_k}$ energy for removing part i where f_1, f_2, \dots, f_k are the parts that are directly connected to the i -th and haven't been removed.

Help the child to find out, what is the minimum total energy he should spend to remove all n parts.

No Dia das Crianças, a criança ganha um brinquedo from Delayyy como presente. Contudo, a criança é tão levada que ela não pode esperar para destruir o brinquedo.

O brinquedo é formado por n partes e m cordas. Cada corda une duas partes, mas cada par de partes é unido por, no máximo, uma corda. Para quebrar o brinquedo, a criança deve remover todas as suas partes. A criança pode remover uma única parte por vez, e cada remoção consome energia. Vamos definir a energia da parte i como v_i . A criança gasta energia igual a $v_{f_1} + v_{f_2} + \dots + v_{f_k}$ para remover a parte i , onde f_1, f_2, \dots, f_k são as partes que estão diretamente conectadas à i -ésima parte e que não foram removidas ainda.

Auxilie a criança a descobrir qual é o total mínimo de energia que ela deve gastar para remover todas as n partes.

Input

The first line contains two integers n and m ($1 \leq n \leq 1000$; $0 \leq m \leq 2000$). The second line contains n integers: v_1, v_2, \dots, v_n ($0 \leq v_i \leq 10^5$). Then followed m lines, each line contains two integers x_i and y_i , representing a rope from part x_i to part y_i ($1 \leq x_i, y_i \leq n$; $x_i \neq y_i$).

Consider all the parts are numbered from 1 to n .

Output

Output the minimum total energy the child should spend to remove all n parts of the toy.

Entrada

A primeira linha contém dois inteiros n e m ($1 \leq n \leq 1000; 0 \leq m \leq 2000$). A segunda linha contém n inteiros: v_1, v_2, \dots, v_n ($0 \leq v_i \leq 10^5$). Então seguem m linhas, e cada linha contém dois inteiros x_i e y_i , representando uma corda da parte x_i para a parte y_i ($1 \leq x_i, y_i \leq n; x_i \neq y_i$).

Considere que todas as partes são numeradas de 1 a n .

Saída

Imprima o total mínimo de energia que a criança deveria gastar para remover todas as n partes do brinquedo.

Exemplo de entrada e saída

Exemplo de entrada e saída

4 3

Exemplo de entrada e saída

4 3



número de partes

Exemplo de entrada e saída

4 3



número de cordas

Exemplo de entrada e saída

4 3

1

2

3

4

Exemplo de entrada e saída

4 3

10 20 30 40

1

2

3

4

Exemplo de entrada e saída

4 3
10 20 30 40
↓
 v_1

1

2

3

4

Exemplo de entrada e saída

4 3
10 20 30 40
↓
 v_2

1

2

3

4

Exemplo de entrada e saída

4 3
10 20 30 40
↓
 v_3

1

2

3

4

Exemplo de entrada e saída

4 3
10 20 30 40
↓
 v_4

1

2

3

4

Exemplo de entrada e saída

4 3

10 20 30 40



Exemplo de entrada e saída

4 3
10 20 30 40
1 4

1₁₀

2₂₀

3₃₀

4₄₀

Exemplo de entrada e saída

4 3
10 20 30 40
1 4
↓
corda

1₁₀

2₂₀

3₃₀

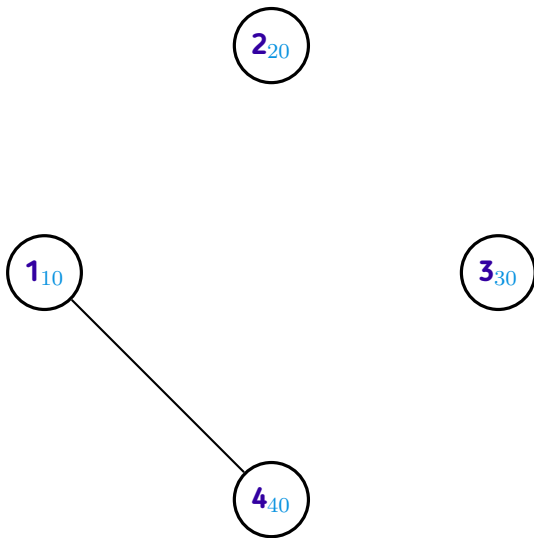
4₄₀

Exemplo de entrada e saída

4 3

10 20 30 40

1 4



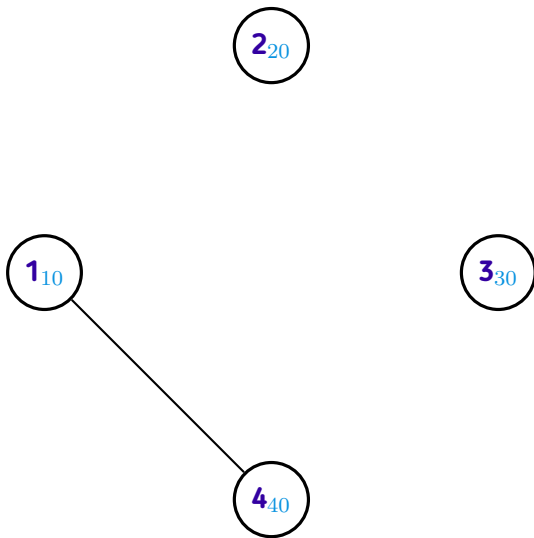
Exemplo de entrada e saída

4 3

10 20 30 40

1 4

1 2



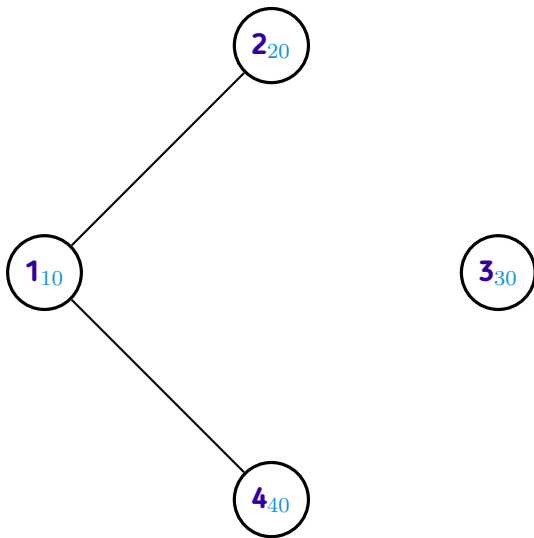
Exemplo de entrada e saída

4 3

10 20 30 40

1 4

1 2



Exemplo de entrada e saída

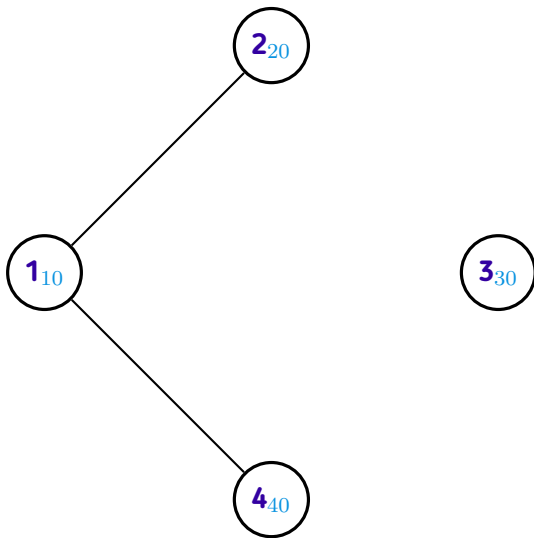
4 3

10 20 30 40

1 4

1 2

2 3



Exemplo de entrada e saída

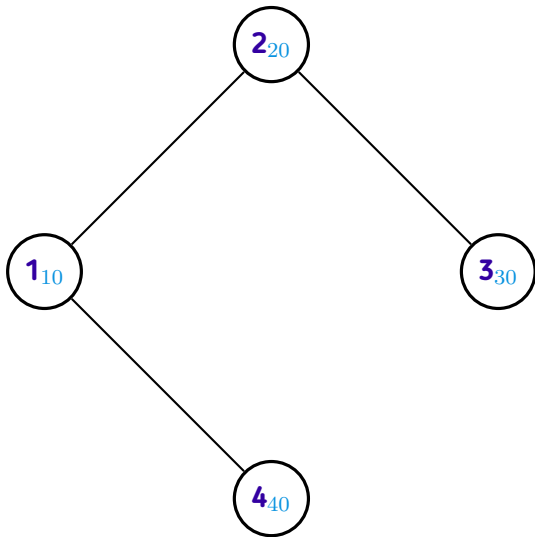
4 3

10 20 30 40

1 4

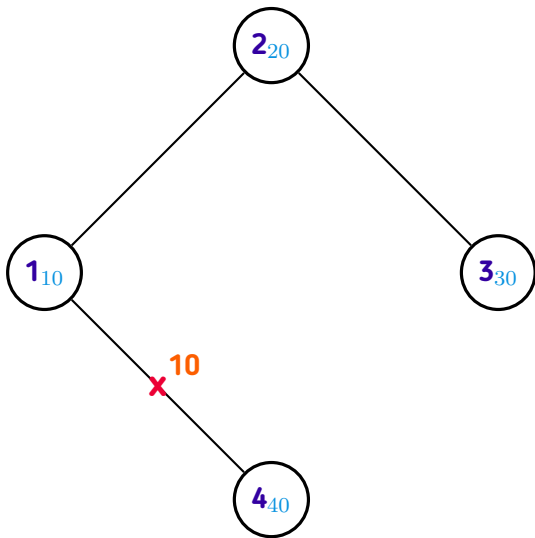
1 2

2 3



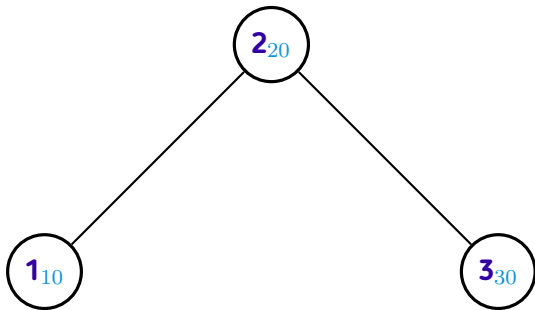
Exemplo de entrada e saída

4 3
10 20 30 40
1 4
1 2
2 3



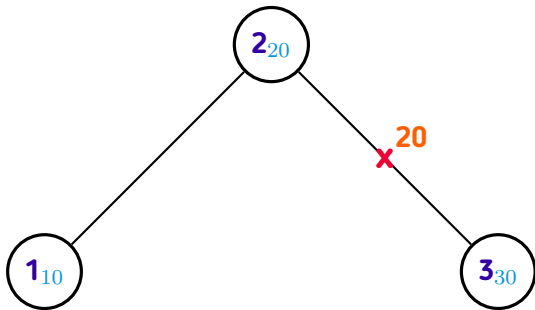
Exemplo de entrada e saída

4 3
10 20 30 40
1 4
1 2
2 3



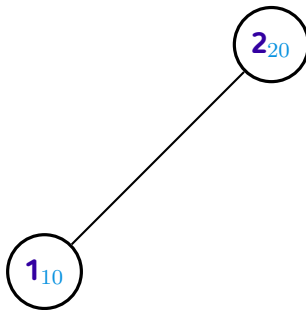
Exemplo de entrada e saída

4 3
10 20 30 40
1 4
1 2
2 3



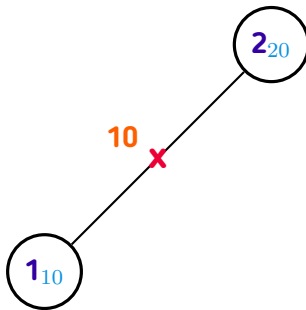
Exemplo de entrada e saída

4 3
10 20 30 40
1 4
1 2
2 3



Exemplo de entrada e saída

4 3
10 20 30 40
1 4
1 2
2 3



Exemplo de entrada e saída

4 3

10 20 30 40

1 4

1 2

2 3

1₁₀

Exemplo de entrada e saída

4 3

10 20 30 40

1 4

1 2

2 3

Exemplo de entrada e saída

4 3

10 20 30 40

1 4

1 2

2 3



40

Exemplo de entrada e saída

Exemplo de entrada e saída

7 10

Exemplo de entrada e saída

7 10

1

7

2

3

4

6

5

Exemplo de entrada e saída

7 10

10 20 30 40

1

7

2

3

4

6

5

Solução

Solução

- ★ Cada viagem consiste em um problema de caminho mínimo

Solução

- ★ Cada viagem consiste em um problema de caminho mínimo
- ★ As arestas tem peso 1, exceto a direção da corrente, que tem peso 0

Solução

- ★ Cada viagem consiste em um problema de caminho mínimo
- ★ As arestas tem peso 1, exceto a direção da corrente, que tem peso 0
- ★ Portanto o problema pode ser resolvido por meio de uma BFS 0/1

Solução

- ★ Cada viagem consiste em um problema de caminho mínimo
- ★ As arestas tem peso 1, exceto a direção da corrente, que tem peso 0
- ★ Portanto o problema pode ser resolvido por meio de uma BFS 0/1
- ★ Complexidade: $O(nrc)$


```
#define REPN(i, N) for (int (i) = 0; (i) < N; (i)++)
```

```
const int MAX { 1010 };
```

```
int vs[MAX];
```

```
vector<int> adj[MAX];
```

```
bitset<MAX> deleted;
```

```
void solve(int N)
```

```
{
```

```
    priority_queue<ii> pq;
```

```
    REP(u, 1, N)
```

```
        pq.push(ii(vs[u], u));
```

```
    auto ans = 0;
```

```
// A melhor estratégia é deletar os vértices em ordem decrescente de custo  
// Prova: cada aresta (u, v) contribui com vs[u] ou vs[v] ao final. Se o maior for  
// excluído, a aresta contribui apenas com min(vs[u], vs[v])
```

```
{  
    auto p = pq.top();  
    pq.pop();  
  
    int u = p.second;  
    int cost = 0;  
  
    for (const auto& v : adj[u])  
        if (not deleted[v])  
            cost += vs[v];  
  
    ans += cost;  
    deleted[u] = true;  
}  
  
cout << ans << endl;  
}  
  
int main()  
{
```