Codeforces Sponsored by Telegram

Codeforces Round #364 (Div. 1)

A. As Fast As Possible

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

On vacations n pupils decided to go on excursion and gather all together. They need to overcome the path with the length l meters. Each of the pupils will go with the speed equal to v_1. To get to the excursion quickly, it was decided to rent a bus, which has seats for k people (it means that it can't fit more than k people at the same time) and the speed equal to v_2. In order to avoid seasick, each of the pupils want to get into the bus **no more than once**.

Determine the minimum time required for all n pupils to reach the place of excursion. Consider that the embarkation and disembarkation of passengers, as well as the reversal of the bus, take place immediately and this time can be neglected.

Input

The first line of the input contains five positive integers n, l, v_1, v_2 and k ($1 \le n \le 10\,000$, $1 \le l \le 10^9$, $1 \le v_1 < v_2 \le 10^9$, $1 \le k \le n$) — the number of pupils, the distance from meeting to the place of excursion, the speed of each pupil, the speed of bus and the number of seats in the bus.

Output

Print the real number — the minimum time in which all pupils can reach the place of excursion. Your answer will be considered correct if its absolute or relative error won't exceed 10^{-6}.

Examples

input

```
5 10 1 2 5
```

output

```
5.0000000000
```

input

```
3 6 1 2 1
```

output

```
4.7142857143
```

Note

In the first sample we should immediately put all five pupils to the bus. The speed of the bus equals 2 and the distance is equal to 10, so the pupils will reach the place of excursion in time $10 / 2 = 5$.

B. Connecting Universities

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Treeland is a country in which there are n towns connected by $n - 1$ two-way road such that it's possible to get from any town to any other town.

In Treeland there are $2k$ universities which are located in different towns.

Recently, the president signed the decree to connect universities by high-speed network.The Ministry of Education understood the decree in its own way and decided that it was enough to connect each university with another one by using a cable. Formally, the decree will be done!

To have the maximum sum in the budget, the Ministry decided to divide universities into pairs so that the total length of the required cable will be maximum. In other words, the total distance between universities in k pairs should be as large as possible.

Help the Ministry to find the maximum total distance. Of course, each university should be present in only one pair. Consider that all roads have the same length which is equal to 1.

Input

The first line of the input contains two integers n and k ($2 \le n \le 200\,000$, $1 \le k \le n / 2$) — the number of towns in Treeland and the number of university pairs. Consider that towns are numbered from 1 to n.

The second line contains $2k$ distinct integers $u_1, u_2, ..., u_{2k}$ ($1 \le u_i \le n$) — indices of towns in which universities are located.

The next $n - 1$ line contains the description of roads. Each line contains the pair of integers x_j and y_j ($1 \le x_j, y_j \le n$), which means that the j-th road connects towns x_j and y_j. All of them are two-way roads. You can move from any town to any other using only these roads.

Output

Print the maximum possible sum of distances in the division of universities into k pairs.

Examples

input

```
7 2
1 5 6 2
1 3
3 2
4 5
3 7
4 3
4 6
```

output

```
6
```

input

```
9 3
3 2 1 6 5 9
8 9
3 2
2 7
3 4
7 6
4 5
2 1
2 8
```

output

```
9
```

Note

The figure below shows one of possible division into pairs in the first test. If you connect universities number 1 and 6 (marked in red) and universities number 2 and 5 (marked in blue) by using the cable, the total distance will equal 6 which will be the maximum sum in this example.

C. Break Up

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Again, there are hard times in Berland! Many towns have such tensions that even civil war is possible.

There are n towns in Reberland, some pairs of which connected by two-way roads. It is not guaranteed that it is possible to reach one town from any other town using these roads.

Towns s and t announce the final break of any relationship and intend to rule out the possibility of moving between them by the roads. Now possibly it is needed to close several roads so that moving from s to t using roads becomes impossible. Each town agrees to spend money on closing no more than one road, therefore, the total number of closed roads will be **no more than two**.

Help them find set of no more than two roads such that there will be no way between s and t after closing these roads. For each road the budget required for its closure was estimated. Among all sets find such that the total budget for the closure of a set of roads is minimum.

Input

The first line of the input contains two integers n and m ($2 \le n \le 1000$, $0 \le m \le 30\,000$) — the number of towns in Berland and the number of roads.

The second line contains integers s and t ($1 \le s, t \le n$, $s \ne t$) — indices of towns which break up the relationships.

Then follow m lines, each of them contains three integers x_i, y_i and w_i ($1 \le x_i, y_i \le n$, $1 \le w_i \le 10^9$) — indices of towns connected by the i-th road, and the budget on its closure.

All roads are bidirectional. It is allowed that the pair of towns is connected by more than one road. Roads that connect the city to itself are allowed.

Output

In the first line print the minimum budget required to break up the relations between s and t, if it is allowed to close no more than two roads.

In the second line print the value c ($0 \le c \le 2$) — the number of roads to be closed in the found solution.

In the third line print in any order c diverse integers from 1 to m — indices of closed roads. Consider that the roads are numbered from 1 to m in the order they appear in the input.

If it is impossible to make towns s and t disconnected by removing no more than 2 roads, the output should contain a single line `-1`.

If there are several possible answers, you may print any of them.

Examples

input

```
6 7
1 6
2 1 6
2 3 5
3 4 9
4 6 4
4 6 5
4 5 1
3 1 3
```

output

```
8
2
2 7
```

input

```
6 7
1 6
2 3 1
1 2 2
1 3 3
4 5 4
3 6 5
4 6 6
1 5 7
```

output

```
9
2
4 5
```

input

```
5 4
1 5
2 1 3
```

```
3 2 1
3 4 4
4 5 2
```

output

```
1
1
2
```

input

```
2 3
1 2
1 2 734458840
1 2 817380027
1 2 304764803
```

output

```
-1
```

D. Huffman Coding on Segment

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alice wants to send an important message to Bob. Message $a = (a_1, ..., a_n)$ is a sequence of positive integers (*characters*).

To compress the message Alice wants to use binary Huffman coding. We recall that *binary Huffman code*, or *binary prefix code* is a function f, that maps each letter that appears in the string to some binary string (that is, string consisting of characters '0' and '1' only) such that for each pair of different characters a_i and a_j string $f(a_i)$ is not a prefix of $f(a_j)$ (and vice versa). The result of the encoding of the message $a_1, a_2, ..., a_n$ is the concatenation of the encoding of each character, that is the string $f(a_1)f(a_2)... f(a_n)$. Huffman codes are very useful, as the compressed message can be easily and uniquely decompressed, if the function f is given. Code is usually chosen in order to minimize the total length of the compressed message, i.e. the length of the string $f(a_1)f(a_2)... f(a_n)$.

Because of security issues Alice doesn't want to send the whole message. Instead, she picks some substrings of the message and wants to send them separately. For each of the given substrings $a_{l_i}... a_{r_i}$ she wants to know the minimum possible length of the Huffman coding. Help her solve this problem.

Input

The first line of the input contains the single integer n $(1 \le n \le 100\,000)$ — the length of the initial message. The second line contains n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 100\,000)$ — characters of the message.

Next line contains the single integer q $(1 \le q \le 100\,000)$ — the number of queries.

Then follow q lines with queries descriptions. The i-th of these lines contains two integers l_i and r_i $(1 \le l_i \le r_i \le n)$ — the position of the left and right ends of the i-th substring respectively. Positions are numbered from 1. Substrings may overlap in any way. The same substring may appear in the input more than once.

Output

Print q lines. Each line should contain a single integer — the minimum possible length of the Huffman encoding of the substring $a_{l_i}... a_{r_i}$.

Example

input

```
7
1 2 1 3 1 2 1
5
1 7
1 3
3 5
2 4
4 4
```

output

```
10
3
3
5
0
```

Note

In the first query, one of the optimal ways to encode the substring is to map 1 to "0", 2 to "10" and 3 to "11".

Note that it is correct to map the letter to the empty substring (as in the fifth query from the sample).

E. Cool Slogans

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Bomboslav set up a branding agency and now helps companies to create new logos and advertising slogans. In term of this problems, *slogan* of the company should be a non-empty substring of its name. For example, if the company name is "hornsandhoofs", then substrings "sand" and "hor" could be its slogans, while strings "e" and "hornss" can not.

Sometimes the company performs rebranding and changes its slogan. Slogan A is considered to be *cooler* than slogan B if B appears in A as a substring **at least twice** (this occurrences are allowed to overlap). For example, slogan $A =$ "abacaba" is cooler than slogan $B =$ "ba", slogan $A =$ "abcbcbe" is cooler than slogan $B =$ "bcb", but slogan $A =$ "aaaaaa" is not cooler than slogan $B =$ "aba".

You are given the company name w and your task is to help Bomboslav determine the length of the longest sequence of slogans $s_1, s_2, ..., s_k$, such that any slogan in the sequence is cooler than the previous one.

Input

The first line of the input contains a single integer n $(1 \le n \le 200\ 000)$ — the length of the company name that asks Bomboslav to help. The second line contains the string w of length n, that consists of lowercase English letters.

Output

Print a single integer — the maximum possible length of the sequence of slogans of the company named w, such that any slogan in the sequence (except the first one) is cooler than the previous

Examples

input

```
3
abc
```

output

```
1
```

input

```
5
ddddd
```

output

```
5
```

input

```
11
abracadabra
```

output

```
3
```