Problem B. Basic Graph Algorithm

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Aqua is learning graph theory. After learning about Depth-First Search, he wrote the following pseudo-code:

```
function dfs(u)
    mark u as visited
    output u
    for v in vertices adjacent to u
        if v is not visited, then
            dfs(v)
        Endif
    Endfor
Endfunction

function run_dfs()
    for v in all vertices
        if v is not visited, then
            dfs(v)
        Endif
    Endfor
Endfunction
```

Aqua noticed that there are some non-deterministic behaviors in line 4 and line 12. Specifically, the order of the traversed vertices might not be fixed!

Therefore, Aqua asks you the following problem: Given an undirected graph with n vertices and m edges and a permutation $p_1, p_2, \ldots, p_n$, what is the minimum number of edges you need to add to **make it possible** for the output order to become the permutation p? Moreover, you need to provide the added edges for Aqua to check your answer.

It can be shown that there always exists an answer to Aqua's problem.

Input

The first line of the input contains two integers n, m ($1 \leq n \leq 3 \times 10^5, 0 \leq m \leq 5 \times 10^5$), indicating the number of vertices and the number of edges.

m lines follow, and the i-th of which contains two integers u_i, v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$), indicating that the i-th edge connects vertex u_i and vertex v_i.

The last line contains n integer $p_1, p_2, \ldots, p_n$, indicating the given permutation.

Output

Output an integer k, the minimum number of edges you need to add, in the first line.

k lines follow, and the i-th of which contains two integers a_i, b_i, indicating the i-th edge you want to add which connects $a_i, b_i (1 \leq a_i, b_i \leq n, a_i \neq b_i)$.

If there are multiple answers, you can output any.

Examples

standard input	standard output
6 6 1 3 1 4 2 3 3 4 3 6 5 6 1 2 3 4 5 6	2 1 2 4 5
8 8 2 8 3 8 5 6 1 6 6 3 8 7 2 3 4 3 1 8 7 5 4 2 3 6	4 1 8 7 5 5 4 4 2