

Min-hashing

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 1024 megabytes

Consider an undirected simple graph $G = (V, E)$. The problem of finding a node with *similar connectivity* is a well-researched topic, because it acts as a good metric to determine which nodes are relevant to other nodes. Services such as "friend recommendation" in Facebook is a good example of its applications. To formalize the notion of similarity, the concept of *Jaccard similarity* can be used, which is defined as $|N(v_1) \cap N(v_2)| / |N(v_1) \cup N(v_2)|$, where $N(v) = \{u \mid (u, v) \in E\}$.

Here, we will instead discuss the *min-hashing* method. Assume each node v has the label l_v . The *shingle value* s_v of node v is defined as $s_v = \min\{l_u \mid u \in N(v)\}$. This method is efficient enough to keep up with industrial needs, and it is also a great metric for similarity: the Jaccard similarity between the set of neighbors $N(v_1)$ and $N(v_2)$ is an unbiased estimator of the probability that nodes v_1 and v_2 have the same shingle values, for random unique labels.

Let's think about a variant of min-hashing: we repeatedly perform min-hashing by taking the label as the previous iteration's shingle value. In this variant, for each node v and the number of iterations k , the value $h_v^{(k)}$ is defined as

$$h_v^{(k)} = \begin{cases} s_v, & \text{if } k = 1 \\ \min\{h_u^{(k-1)} \mid u \in N(v)\}, & \text{if } k \geq 2 \end{cases}$$

For each k , let c_k be the number of unordered pairs of distinct vertices $\{u, v\}$ such that $h_u^{(k)} = h_v^{(k)}$. Then, how does the value c_k change as k increases? In this problem, your task is to compute $\max_{k \in \mathbb{N}} c_k$.

Input

The first line contains two positive integers n and m ($1 \leq n \leq 100\,000, 1 \leq m \leq 250\,000$) representing the number of nodes and the number of edges, respectively. The nodes are numbered from 1 to n . Note that these are **not** the labels of the nodes.

The second line contains n integers comprising a permutation of the first n positive integers, where the i -th number in the line represents the initial label of node i .

Each of the next m lines contains two integers. The i -th of these lines contains two distinct integers u_i and v_i ($1 \leq u_i, v_i \leq n$), which means $\{u_i, v_i\} \in E$.

The input will be set in a way such that there are no self-loops, parallel edges, or nodes with a degree of zero.

Output

Print the maximum value of c_k over all positive integers k .

Examples

standard input	standard output
5 5 1 2 3 4 5 1 2 2 3 3 4 4 5 5 1	10
4 3 1 2 3 4 1 2 2 3 3 4	2