Problem B
Dynamic Graph Queries

Time limit: 1 second
Memory limit: 2048 megabytes

Problem Description

You are given a graph of n verticies, where the vertices are numbered from 1 to n. Each vertex has a color of black or white. There are no edges between the vertices initially.

q queries are given. Each query belongs to one of the following three types:

1. $1\ u_i\ v_i$ —Add an **undirected** edge between vertex u_i and v_i.
2. $2\ u_i$ —Find the number of black verticies in the same connected component as u_i.
3. $3\ u_i$ —Find the nearest distance of a black vertex from vertex u_i. The distance between two verticies is the smallest number of edges in a path between two vertices. If there are no black verticies reachable from vertex u_i, output -1. **Note that the number of this type of query is limited to 100 in this problem.**

Can you answer all these queries?

Input Format

The first line of the input contains two integers n and q. The second line of the input contains n space-separated integers $a_1, a_2, \ldots, a_n$. The i-th vertex is black if $a_i = 1$, otherwise it is white. The i-th of the following q lines contains the i-th query in the format as in the problem description.

Output Format

For each query of type 2 and 3, output the answer to the query in one line.

Technical Specification

- $1 \leq n, q \leq 2 \times 10^5$
- $0 \leq a_i \leq 1$ for $i = 1, 2, \ldots, n$
- $1 \leq u_i < v_i \leq n$ for $i = 1, 2, \ldots, q$ of query type 1
- $1 \leq u_i \leq n$ for $i = 1, 2, \ldots, q$ of query type 2 and 3
- It is guaranteed that the number of queries of type 3 is at most 100. Note that there is no limitation on query of type 2.
- It is guaranteed that there is at least one query of type 2 or 3 in the input.

Scoring

1. (40 points) Only queries of type 1 and 2 appear in the input.
2. (40 points) Only queries of type 1 and 3 appear in the input.
3. (20 points) No additional constraints.

Sample Input 1

```
5 8
1 0 0 1 1
2 4
3 2
1 2 4
3 2
1 1 5
2 5
1 4 5
2 2
```

Sample Output 1

```
1
-1
1
2
3
```

Sample Input 2

```
5 8
0 1 0 0 0
1 2 3
1 3 5
1 1 5
2 1
2 2
2 3
2 4
2 5
```

Sample Output 2

```
1
1
1
0
1
```

Sample Input 3

```
5 7
0 1 0 1 1
1 2 3
1 4 5
3 1
3 2
3 3
3 4
3 5
```

Sample Output 3

```
-1
0
1
0
0
```

Hint

Use a disjoint set data structure to handle queries of type 2. You also need to run BFS to handle queries of type 3.