Hooliganism

Input File: *standard input*
Output File: *standard output*

The City of Sydney has been having problems with "hooliganism" as of late. Out of control youths are driving around non-stop all night, scaring hippos and driving over daisies. Now you must help implement a plan to lock out these hooligans once and for all.

The City of Sydney is made up of N intersections, numbered 1 to N, connected by R roads. Each night all roads in Sydney will become one-way. You decide which way each road goes. To prevent congestion, each intersection has a maximum number of roads that are allowed to leave from it.

You must find a way to make each of these roads one-way such that each intersection has no more than the specified number of roads leaving it, and that there is no way for hooligans to drive around all night. In particular, hooligans should not be able to drive through the same intersection more than once.

Input

- The first line of input will contain two space-separated integers N R, representing the number of intersections and roads in Sydney respectively.
- The following N lines of input will each contain an integer a_i, the i-th of which gives the maximum number of one-way roads allowed to leave intersection i.
- The following R lines of input will each contain two space-separated integers b c, indicating that intersections b and c have a road between them.

Output

If Sydney cannot prevent hooligans from driving around all night, then output `IMPOSSIBLE`.

If it is possible for Sydney to construct such a road network, then output R lines, each containing two integers b_i c_i, indicating a one-way road from intersection b_i to intersection c_i. These lines must describe every road in Sydney, and can be output in any order.

Sample Input 1

```
3 3
0
2
0
1 2
1 3
2 3
```

Sample Output 1

```
IMPOSSIBLE
```

Explanation 1

Intersections 1 and 3 are connected by a road, however neither of them are allowed to have any roads leaving them. Thus it is impossible to assign directions to the roads to satisfy the restrictions on the number of roads leaving each intersection.

Sample Input 2

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

Sample Output 2

```
IMPOSSIBLE
```

Explanation 2

Here it is possible to assign directions to the roads, however it requires you to make a ring going from intersection 1 to 2 to 3 and back to 1, or the opposite direction. This would allow hooligans to drive past the same location multiple times, so this case is also impossible.

Sample Input 3

```
3 3
1
2
0
1 2
1 3
2 3
```

Sample Output 3

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

Explanation 3

Once hooligans enter intersection 3, they cannot leave. The only place for hooligans to go from intersection 1 is intersection 3, thus once a hooligan arrives at intersection 1 they are no longer trouble. Similarly, a hooligan at intersection 2 can go to either intersection 1 or intersection 3, but either way they arrive at intersection 3 and cannot go any further. Thus no hooligan can drive past the same location more than once.

Subtasks & Constraints

For all subtasks, $1 \leq R \leq 100\,000$ and $2 \leq N \leq 100\,000$. Additionally, no two intersections will be directly connected by more than one road, and no intersection will be connected to itself. It is not necessarily possible to go from any intersection to any other.

- For Subtask 1 (20 points), $a_i = N$, that is, the maximum number of roads allowed to leave each intersection is the same as the number of intersections.
- For Subtask 2 (20 points), every intersection has exactly two other intersections connected to it.
- For Subtask 3 (30 points), $1 \leq R \leq 100\,000$ and $2 \leq N \leq 1\,000$.
- For Subtask 4 (30 points), no further constraints apply.

Scoring

The score for each input scenario will be 100% if a correct answer is written to the output file, and 0% otherwise.