

Reverse an Edge

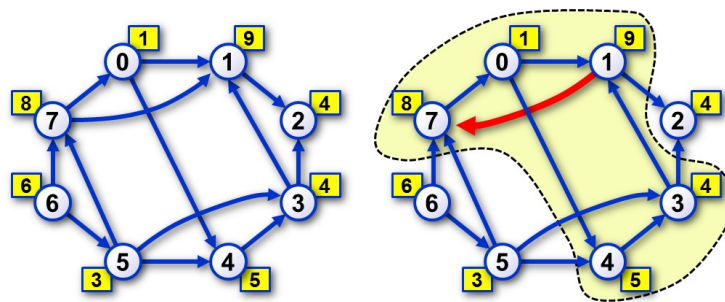


Image 1. Example of a directed acyclic graph with eight weighted nodes labeled 0, 1, ..., 7. Each weight is depicted in the rectangle at the corresponding node. By reversing the edge (7, 1) a strongly connected component is created, it is highlighted on the right side of the image. The nodes of the component are 0, 1, 3, 4 and 7, the sum of weights of these nodes is equal to 27.

The Task

You are given a directed acyclic graph with positive node weights. Your task is to choose exactly one edge whose reversal will create a strongly connected component with maximum possible sum of values of its nodes.

Input

The first line contains two integers N , M separated by space and representing the number of nodes and the number of edges in the graph. The nodes are labeled 0, 1, 2, ..., $N-1$. Next, there are N lines, the i -th line contains the weight of the node labeled by $i-1$ (The first line corresponds to node 0, the second line corresponds to node 1, etc). All weights are positive integers. Next, there are M lines, each line represents one edge. The edge is specified by the labels of its endnodes, the labels are separated by space and the label of the source node of the edge is listed first. The list of all edges is presented in an arbitrary order.

The value of N does not exceed 5000, the value of $N \times M$ does not exceed 2.5×10^8 .

Output

The output is one test line with three integers A , B , C , separated by space. The values A and B are the labels of the end nodes of the desired edge before its reversal. The original edge direction is $A \rightarrow B$. The value C represents the sum of node values in the strongly connected component created by reversing the direction of (A, B) . When more than one edge fulfills the conditions of the task, the output lists only one of them, namely that one whose notation is lexicographically minimal when represented as an integer tuple (edge_source_node, edge_target_node).

It is guaranteed that the resulting strongly connected component contains at least three nodes. The sum of the node values in the component does not exceed 10^9 .

Example 1

Input

```
8 13
1
9
4
4
5
3
6
8
0 1
1 2
3 2
4 3
5 4
6 5
6 7
7 0
7 1
3 1
5 3
5 7
0 4
```

Output

```
7 1 27
```

The graph in the Example 1 is depicted in the Image 1.

Example 2

Input

```
8 9
5
5
20
7
5
5
7
7
2 1
1 0
1 5
0 4
4 5
2 3
3 7
6 3
6 7
```

Output

```
6 7 21
```

Example 3

Input

```
5 10
10
20
30
40
50
0 1
0 2
0 3
0 4
1 2
1 3
1 4
2 3
2 4
3 4
```

Output

```
0 4 150
```

Public data

The public data set is intended for easier debugging and approximate program correctness checking. The public data set is stored also in the upload system and each time a student submits a solution it is run on the public dataset and the program output to stdout and stderr is available to him/her.

[Link to public data set](#)