Task: DMC
Dexter's Mouse Chase

AACPP WiSe 2025/26 Round 4 Memory: 64MiB 2025.12.03 – 2025.12.17

Dexter the Cat has finally detected the secret route that the Big Mouse uses to travel from its burrow to its hidden cheese vault in the countryside. Dexter knows when the mouse will travel and he knows that it always travels through the **Forest Trail Network**.

The Forest Trail Network consists of two-way walking paths, each directly connecting two distinct trail markers. A trail marker may connect to any number of other markers. Dexter knows these facts:

- The mouse will enter the trail system at the trail marker a closest to its burrow.
- It will leave the system at the trail marker b closest to its cheese vault.
- It will not leave and re-enter the trails in between.

Dexter wants to set up **mouse traps** at some of the trail markers. When the mouse passes through a marker containing a trap, Dexter can finally catch it. Each trap has a certain deployment cost (Dexter needs to travel there in advance and install it in a suitable place).

Dexter wants to *minimise the total deployment cost* while ensuring that the mouse cannot reach the exit from the entry without going through at least one trapped marker.

Your task is to compute such a minimum-cost **controlling set** of trail markers. You may assume that at least one valid route exists between the mouse's entry and exit markers.

Input

The first line of standard input contains two integers n and m – the number of trail markers and the number of paths between them. The markers are numbered from 1 to n.

The second line contains two integers a and b, $1 \leq a, b \leq n, a \neq b$ – the numbers of the markers closest to the burrow and to the vault, respectively.

Next n lines describe the costs of setting up traps. The i-th of these lines contains one integer w_i – the cost of setting up a trap at the i-th marker.

Finally, the m following lines describe the Forest Trail Network. In each line there are two integers x_i and y_i, $1 \leq x_i, y_i \leq n, x_i \neq y_i$, indicating that there is a two-way trail connecting markers x_i and y_i. The trail pairs do not repeat.

Output

The first line of output should contain the minimal cost of a controlling set.

The second line of output should contain the numbers of trail markers in the controlling set **in ascending order**. If there exists more than one controlling set, your program may output any one of them.

Example

For the input:

```
5 6
5 3
2
4
8
3
10
1 5
1 2
2 4
4 5
2 3
3 4
```

the correct output is:

```
5
1 4
```

Additional examples

The following initial tests are also available:

- `0b` – $n = 20$, $m = 19$, $a = 1$, $b = 20$, the trail is a path $1 \to 2 \to ... \to 19 \to 20$; each mouse trap costs 50 except for the one at trail marker 11, which costs 25; the optimal solution is to trap marker 11.
- `0c` – $n = 200$, $m = n$, $w_i = (i \bmod n) + 1$, $a = 50$, $b = 150$, the trail is a cycle $1 \to 2 \to ... \to 199 \to 200 \to 1$. The optimal solution is to trap markers 100 and 200 for a total cost of 2.
- `0d` – $n = 200$, $m = 19\,900$, $a = 37$, $b = 113$, all costs are 10^7, there exists a trail between each pair of markers; there are two optimal solutions: trapping a or trapping b, both with cost 10^7.

Limits

Your solution will be evaluated on a number of hidden test cases divided into groups. Points for a group are awarded if and only if the submission returns the correct answer for each of the tests in the group within the allotted time limit. These groups are organised into subtasks with the following limits and points awarded.

Subtask	Limits	Points
1.	$2 \leq n \leq 20$, $1 \leq m \leq 100$, $w_i \leq 100$	2
2.	$2 \leq n \leq 200$, $m \leq 19\,900$, $w_i \leq 100$	4
3.	$2 \leq n \leq 200$, $m \leq 19\,900$, $w_i \leq 10^7$	4

Partial points

If your solution outputs the correct number of radars (first line of output), and the second line is left blank or not correct, it will receive 50% of the points for a given test group.