Magic Stones

Problem

Given a positive integer N, there are N cells in a row, each one unit away from its adjacent cells, numbered from 1 to N. Additionally, there are 2 magic stones, initially located in cells A and B. There are Q queries, and after each query, there must be at least one magic stone in a specific position. Moving a magic stone from position i to position j costs $|i - j|$ units. Find the minimum number of cost units needed to satisfy the Q queries, in the given order.

Implementation Details

You must implement the function *Piedras_Magicas()*. This function receives 4 integers, N, A, B, and Q; the number of cells, the initial positions of the magic stones, and the number of queries, respectively. Additionally, it receives a vector a, with Q elements. $a[i]$ indicates the position where there must be a magic stone at moment i. The function must return a single integer, the minimum number of units needed to satisfy all queries. A sample program would look like this:

```
#include <bits/stdc++.h>
using namespace std;

long long int Piedras_Magicas(int N, int A, int B, int Q, vector<int> a) {
    // Implement this function.
}
```

The grader will call the function **multiple** times for each test case.

Examples

Example 1:

- The grader calls the function

$$Piedras\_Magicas(10,\ 1,\ 10,\ 3,\ \{3,\ 6,\ 1\})$$

- The function should return 8. The queries can be processed as follows:
  - Move the stone from position 1 to position 3.
  - Move the stone from position 10 to position 6.
  - Move the stone from position 3 to position 1.

Example 2:

- The grader calls the function

Piedras_Magicas(11, 1, 11, 2, {6, 1})

- The function should return 5. The queries can be processed as follows:
  - Move the stone from position 11 to position 6.

Example 3:

- The grader calls the function

Piedras_Magicas(11, 1, 11, 2, {6, 11})

- The function should return 5. The queries can be processed as follows:
  - Move the stone from position 1 to position 6.

Example 4:

- The grader calls the function

Piedras_Magicas(11, 8, 1, 16, {1, 1, 5, 1, 11, 4, 5, 2, 5, 3, 3, 3, 5, 5, 6, 7})

- The function should return 21.

Constraints

- $1 \leq N, Q \leq 2 \times 10^5$.
- $1 \leq A, B \leq N$.
- For all $0 \leq i \leq Q - 1$, $1 \leq a[i] \leq N$.
- Let S_N be the sum of the values of N over all function calls. It is guaranteed that $S_N \leq 2 \times 10^5$.
- Let S_Q be the sum of the values of Q over all function calls. It is guaranteed that $S_Q \leq 2 \times 10^5$.

Subtasks

- (10 points) $Q, S_Q \leq 20$.
- (20 points) $N, S_N, Q, S_Q \leq 250$.
- (30 points) $N, S_N, Q, S_Q \leq 2000$.
- (40 points) No additional constraints.