Beautiful Buildings 2

Time and memory limits: 2 seconds, 512 MB

There are two rows of N buildings on opposite sides of a street. The buildings in each row are numbered 1 to N from left to right, so building i in the first row is directly opposite building i in the second row. The ith building in the first row has height A_i, and the ith building in the second row has height B_i. You believe that buildings directly opposite each other look beautiful when they have similar heights.

Each row contains exactly one building of each height from 1 to N. In other words, both A and B are permutations of $1, 2, \ldots, N$.

The *ugliness* of the buildings is the sum of the absolute differences[1] between the heights of buildings directly opposite each other. The ugliness can be expressed as $|A_1 - B_1| + |A_2 - B_2| + \cdots + |A_N - B_N|$.

You may swap the heights of two buildings in the **first row** at most once. You may also choose not to make a swap. What is the minimum ugliness you can achieve?

Subtasks and constraints

Your program will be graded using many secret tests. Every test follows some rules:

- $1 \leq N \leq 1\,000\,000$.
- $1 \leq A_i, B_i \leq N$ for all i.
- $A_i \neq A_j$ and $B_i \neq B_j$ for all $i \neq j$.

The secret tests are divided into subtasks. Your program must correctly solve **every test** within a subtask to earn the marks for that subtask:

- For Subtask 1 (10 marks), $|A_i - B_i| \leq 1$ for all i.
- For Subtask 2 (25 marks), $N \leq 2000$.
- For Subtask 3 (20 marks), there is at most one index i such that $A_i < B_i$.
- For Subtask 4 (45 marks), no special rules apply.

Input

Your solution must read input and print output.

The input follows a specific format:

- The 1st line contains the integer N.
- The 2nd line contains N integers describing the heights in the first row. The ith of these is A_i.
- The 3rd line contains N integers describing the heights in the second row. The ith of these is B_i.

Output

Your program must output one integer: the minimum ugliness you can achieve.

Note: For students using C, C++ or Java, please note that the answer may exceed the maximum value that can be stored in an `int` *integer type. As such, you should consider using the* `long long` *integer type in C or C++, or the* `long` *integer type in Java. Python users do not need to consider this.*

[1] The absolute value of a number (denoted by vertical bars) is its distance from 0. For example, $|2| = |-2| = 2$. The absolute difference between two numbers is the distance between them. For example, $|5 - 10| = 5$.

Sample input 1

```
4
1 2 3 4
4 3 2 1
```

Sample output 1

```
2
```

Sample input 2

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

Sample output 2

```
0
```

Sample input 3

```
5
1 4 2 5 3
5 3 1 4 2
```

Sample output 3

```
6
```

Explanation

- In the 1st sample case, swap the heights of buildings 1 and 4 in the first row. The heights in that row become $4, 2, 3, 1$, giving ugliness $|4-4| + |2-3| + |3-2| + |1-1| = 2$.
- In the 2nd sample case, buildings directly opposite each other already have the same height. Choose not to swap any heights, giving ugliness 0.
- In the 3rd sample case, swap the heights of buildings 1 and 4 in the first row. The heights in that row become $5, 4, 2, 1, 3$, giving ugliness $0 + 1 + 1 + 3 + 1 = 6$.