Programming Contest

A programming organization is planning a contest for several programmers, each of which has a certain rating. (The higher the rating, the better the programmer.) Each programmer is paired with another programmer, and the difference between their ratings is referred to as the "bias amount". Given the ratings of all the programmers in the contest, what is the minimum total bias amount that can be achieved by optimally planning the programmer pairs?

****Example****

*n = 4*

*ratings = [4, 2, 5, 1]*

The optimal solution is:

Pair the second programmer (2) with the fourth (1) for a difference of 1.

Pair the first programmer (4) with the third (5) for a difference of 1.

This results in a total bias amount of 1 + 1 = 2.

**Function Description**

Complete the function *minimizeBias* in the editor below.

minimizeBias has the following parameter:

    int *ratings[n]:*  the ratings of each of the programmers

Returns:

    int: the minimum total bias amount that can be achieved in the contest

**Constraints**

1 ≤ *n* ≤ 105

1 ≤ *ratings[i]* ≤ 10^9

*n* is even.

**Input Format For Custom Testing**

The first line contains an integer, *n*, the number of elements in *ratings*.

Each line *i* of the *n* subsequent lines (where *0 ≤ i < n*) contains an integer, *ratings[i]*.

**Sample Case 0**

**Sample Input For Custom Testing**

STDIN     Function

-----     --------

4 → ratings[] size n = 4

1 → ratings = [1, 3, 6, 6]

3

6

6

**Sample Output**

2

**Explanation**

The optimal solution is to pair the first programmer (1) with the second (3) for a difference of 2, and the third programmer (6) with the fourth (6) for a difference of 0. This results in a total bias amount of 2.

**Sample Case 1**

**Sample Input For Custom Testing**

STDIN    Function

-----    --------

6    →   ratings[] size n = 6

2    →   ratings = [2, 4, 5, 3, 7, 8]

4

5

3

7

8

**Sample Output**

3

**Explanation**

The optimal solution is to assign the following pairs: (2,3), (4,5), and (7,8). This results in the least total bias amount, which is 3.

import java.io.\*;

import java.math.\*;

import java.security.\*;

import java.text.\*;

import java.util.\*;

import java.util.concurrent.\*;

import java.util.function.\*;

import java.util.regex.\*;

import java.util.stream.\*;

import static java.util.stream.Collectors.joining;

import static java.util.stream.Collectors.toList;

class Result {

/\*

\* Complete the 'minimizeBias' function below.

\*

\* The function is expected to return an INTEGER.

\* The function accepts INTEGER\_ARRAY ratings as parameter.

\*/

public static int minimizeBias(List<Integer> ratings) {

// Write your code here

}

}

public class Solution {

public static void main(String[] args) throws IOException {

BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));

BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv("OUTPUT\_PATH")));

int ratingsCount = Integer.parseInt(bufferedReader.readLine().trim());

List<Integer> ratings = IntStream.range(0, ratingsCount).mapToObj(i -> {

try {

return bufferedReader.readLine().replaceAll("\\s+$", "");

} catch (IOException ex) {

throw new RuntimeException(ex);

}

})

.map(String::trim)

.map(Integer::parseInt)

.collect(toList());

int result = Result.minimizeBias(ratings);

bufferedWriter.write(String.valueOf(result));

bufferedWriter.newLine();

bufferedReader.close();

bufferedWriter.close();

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*