

Perfect Array

This problem was worth 500 points and was ranked easy. The authors of this problem are Aryan V S and Dhruv Chawla.

Note: GitHub does not support LaTeX in Markdown. If you want a more readable version of the problem, download the PDF file instead.

Statement

Aryan and Dhruv are given an array a of length n . They are allowed to perform the following operations on the array:

- Aryan can select any number of odd indices and add 1 to the values at these indices
- Dhruv can select any number of even indices and add 1 to the values at these indices

They must try to make all elements of the array equal i.e. $a_1 = a_2 = \dots = a_n$ using minimum number of these operations. They think that the problem is very hard for them and ask for your help.

Input Format

The first line contains a single integer n - the size of the array a .

The second line contains n space separated integers - the elements of the array a .

Output Format

The output should contain a single integer - the minimal number of operations required to make all elements of the array equal.

Constraints

$$1 \leq n \leq 2 \cdot 10^5$$

$$-10^9 \leq a_i \leq 10^9$$

Sample Tests

Sample Test 1

Input

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

Output

1 | 7

Solution

Time Complexity: $O(n)$

Space Complexity: $O(n)$

Notice that we are only allowed to increase the values. So, the optimal solution will not include increasing the maximum number(s) m in a because if it did, then we would have to perform more operations to make the smaller elements equal to it.

So, our goal is to make all elements of the array a equal to the maximum number m in a . But, Aryan can do operations only on the odd indices while Dhruv can do operations only on the even indices.

To make the values at odd indices same as the maximum value in the array, we must increment values at some indices as long as the minimum value is not equal to the m . Same applies for even indices.

Solution in C++:

```
1  #include <bits/stdc++.h>
2
3  int main () {
4      std::ios::sync_with_stdio(false);
5      std::cin.tie(nullptr);
6
7      int n;
8      std::cin >> n;
9
10     std::vector<int64_t> a (n);
11     int64_t max = -1e18;
12
13     for (int i = 0; i < n; ++i) {
14         std::cin >> a[i];
15         max = std::max(max, a[i]);
16     }
17
18     if (n == 1) {
19         std::cout << 0 << '\n';
20         return 0;
21     }
22
23     int64_t min_odd = 1e18;
24     int64_t min_even = 1e18;
25     for (int i = 0; i < n; i += 2)
26         min_even = std::min(min_even, a[i]);
27     for (int i = 1; i < n; i += 2)
28         min_odd = std::min(min_odd, a[i]);
29
30     std::cout << 2 * max - min_odd - min_even << '\n';
31 }
```

```
32     return 0;
33 }
```

Solution in Python:

```
1  n = int(input())
2  a = list(map(int, input().split()))
3
4  if n == 1:
5      print(0)
6      exit(0)
7
8  m = max(a)
9  min_even = min(a[0::2])
10 min_odd = min(a[1::2])
11
12 print((m - min_even) + (m - min_odd))
```