Practice(https://practice.geeksforgeeks.org/home/)

Problem Submissions

All Problems (JavaScript:void(0))

Minimum Differencing

Accuracy: 4.51% Submissions: 857 Points: 40

Given an array **arr** of size **N** and an integer **K** which is a **power of 2** . You have to **minimize the difference** of a function by applying the **special operation** any number of times.

Function is given by = **maximum element in arr** - **minimum element in arr.**

A **special operation** has to be performed any number of times.

- If the number is even, divide it by **2.**
- If the number is odd, multiply it by **K (**Note that this operation works only on initially odd element**).**

Example 1 :

```
Input :
3 4
1 2 6

Output :
1

Explaination :
Divide the 3rd element by 2 then elements are 1 2 3.
Multiply 1st element with 4 then elements are 4 2 3.
Divide 1st element by 2 then elements are 2 2 3.
So difference mimimum possible is 1.
```

Example 2 :

```
Input :
2 8
1 8

Output :
0
```

Output Window

C++ (g++ 5.4) Test against custom input

```
// } Driver Code Ends
class Solution {
  public:
    long long minimizeDifference(int n, int k, vector<int> &a) {
        set<long long> se;
        for(int i=0;i<n;i++){
            if(a[i]%2==1){
                long long ele=k;
                se.insert(ele*a[i]);
            }
            else{
                se.insert(a[i]);
            }
        }
        long long mi=*se.begin();
        priority_queue<long long> li;
        for(auto it=se.begin();it!=se.end();it++){
            li.push(*it);
        }
        int length=li.size();
        long long ans=1e14;
        while(li.top()%2==0){
            long long num=li.top();
            li.pop();
            li.push(num/2);
            mi=min(mi,num/2);
            ans=min(ans,li.top()-mi);
        }
        return ans;
    }
};

// } Driver Code Ends
```

Compile & Run Submit