

# PERFECTION

An array of positive integers is called *perfect* if, for every positive integer  $K$ , the value  $K$  appears either **0 times** or **exactly  $K$  times**. An empty array is also considered perfect. For example, the array  $[2, 4, 2, 4, 4, 4]$  is perfect because 2 appears exactly 2 times, 4 appears exactly 4 times, and every other value appears 0 times.

You are given an array of  $N$  positive integers  $V_1, \dots, V_N$  and may perform the following operations any number of times, in any order:

- **Add** an element (any positive integer) for  $A$  coins.
- **Delete** an element for  $D$  coins.
- **Modify** an element (change its value to any positive integer) for  $M$  coins.

What is the minimum number of coins required to make the array perfect?

## Subtasks and Constraints

For all subtasks:

- $1 \leq N \leq 5\,000$ .
- $1 \leq V_i \leq 5\,000$  for all  $i$ .
- $1 \leq A, D, M \leq 1\,000\,000\,000$ .

Additional constraints for each subtask are given below.

Subtask	Points	Additional constraints
1	10	$N = 1$ .
2	10	$V_i = V_j$ for all $i, j$ .
3	15	$M > A + D$ .
4	25	$M = 1, A = 1\,000\,000\,000, D = 1\,000\,000\,000$ .
5	20	$N \leq 300, V_i \leq 300$ for all $i$ .
6	20	No additional constraints.

## Input

- The first line contains four integers  $N, A, D$ , and  $M$ .
- The second line contains  $N$  integers  $V_1, \dots, V_N$ .

## Output

Output a single integer, the minimum number of coins required to make the array perfect.

Note: Your solution may involve integers which are large. Consider using 64-bit integers ('long long' in C++) in your solution.

**Sample Input 1**

```
6 1 2 3
2 4 2 4 4 4
```

**Sample Output 1**

```
0
```

**Sample Input 2**

```
1 5 3 4
2
```

**Sample Output 2**

```
3
```

**Sample Input 3**

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

**Sample Output 3**

```
3
```

**Sample Input 4**

```
14 10 6 5
3 3 3 3 3 4 4 5 5 5 5 5 5 14
```

**Sample Output 4**

```
20
```

**Sample Input 5**

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

**Sample Output 5**

```
7
```

**Explanation**

In the first sample case, the array  $[2, 4, 2, 4, 4, 4]$  is already perfect, so no operations are required. The answer is 0.

In the second sample case, we can delete the element 2 for 3 coins. This results in the empty array, which is perfect. There is no cheaper option, so the answer is 3.

In the third sample case, we can modify one 1 into a 3 for 2 coins and add another 3 for 1 coin. This results in  $[3, 3, 1, 3]$  which is a perfect array. There is no cheaper option, so the answer is 3.

In the fourth sample case, we can modify two 3s into 2s for 10 coins, modify a 5 into a 4 for 5 coins and modify the 14 into a 4 for 5 coins. This results in  $[2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5, 5, 5, 4]$  which is a perfect array. There is no cheaper option, so the answer is 20.

In the fifth sample case, we can modify a 2 into a 1 for 3 coins and delete a 2 for 4 coins. This results in  $[2, 2, 1]$  which is a perfect array. There is no cheaper option, so the answer is 7.