5. Equalizing Array Elements

Given an array of integers, transform it so that at least a certain number of elements in the array are equal. To achieve this, you can perform an operation where you select an element in the array and divide it by the given division parameter using integer division. What is the minimum number of operations that must be performed to achieve this goal on a certain array?

For example, let's say *arr = [1, 2, 3, 4, 5]*. The desired number of equal elements is denoted as *threshold = 3*, and the division parameter is *d = 2*. If you divide the value 4 once and the value 5 once using integer division, you get the array [1, 2, 3, 2, 2], which contains 3 equal elements. There is no way to achieve this in less than 2 operations. Therefore, the answer is 2.

Function Description
Complete the function *minOperations* in the editor below.

minOperations has the following parameter(s):
  int *arr[n]:* an array of integers
  int *threshold:* the minimum number of desired equal elements in the array
  int *d:* the division parameter used to divide an element in a single operation
Returns:
  int: the minimum number of operations required to have at least *threshold* number of equal elements in the array

Constraints

- $1 \le n \le 3*10^4$
- $1 \le arr[i] \le 2*10^5$
- $1 \le threshold \le n$
- $2 \le d \le 1000$

▾ Input Format For Custom Testing

The first line contains an integer, *n*, denoting the size of the array.
Each line *i* of the *n* subsequent lines (where $0 \le i < n$) contains an integer that describes *arr[i]*.
The next line contains an integer, *threshold*, denoting the minimum number of desired equal elements in the array.
The next line contains an integer, *d*, denoting the division parameter.

▾ Sample Case 0

Sample Input For Custom Testing

```
4
64
30
25
33
2
2
```

Sample Output

```
3
```

Explanation
In this case, *arr = [64, 30, 25, 33]*, *threshold = 2*, and the division parameter *d = 2*. In other words, the minimum required number of equal elements is 2, and in one operation we can divide a single element by 2 using integer division. If we divide 64 twice to get 16, and we divide 33 once to also get 16, Then the array becomes [16, 30, 25, 16], which has 2 equal elements. There is no way to get at least 2 equal elements with fewer than 3 operations. Therefore, the answer is 3.

▾ Sample Case 1

Sample Input For Custom Testing

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

Sample Output

```
6
```

Explanation
In this case, the *arr = [1, 2, 3, 4]*, *threshold = 4*, and the division parameter *d = 3*. In other words, the minimum required number of equal elements is 4, and in one operation we can divide a single element by 3 using integer division. The only way to get all 4 elements to be equal is to divide all of them so they all become 0. One operation is required to convert 1 to 0, and another single operation is required to convert 2 to 0. Two operations are required to convert 3 to 0, and another two operations are needed to convert 4 to 0. Therefore, the total number of required operations is 1+1+2+2 = 6.