

Description

Solution

Discuss (774)

Submissions

1356. Sort Integers by The Number of 1 Bits

Easy

915

42

Add to List

Share

You are given an integer array `arr` . Sort the integers in the array in ascending order by the number of `1` 's in their binary representation and in case of two or more integers have the same number of `1` 's you have to sort them in ascending order.

Return *the array after sorting it*.

Example 1:

Input:

arr = [0,1,2,3,4,5,6,7,8]

Output:

[0,1,2,4,8,3,5,6,7]

Explantion:

[0] is the only integer with 0 bits.
[1,2,4,8] all have 1 bit.
[3,5,6] have 2 bits.
[7] has 3 bits.
The sorted array by bits is [0,1,2,4,8,3,5,6,7]

Example 2:

Input:

[1024,512,256,128,64,32,16,8,4,2,1]

Output:

[1,2,4,8,16,32,64,128,256,512,1024]

Explantion:

All integers have 1 bit in the binary representation, you should just sort them in ascending order.

Constraints:

- 1 <= arr.length <= 500
- 0 <= arr[i] <= 10⁴

Accepted 60,318

Submissions 84,887

Seen this question in a real interview before?

Yes

No

Companies

Related Topics

i

Java

Autocomplete

i

{ }

↺

⚙

⌵

```
8         counter++;
9         number = number & (number - 1);
10      }
11
12      return counter;
13
14  }
15
16  public int[] sortByBits( int[] arr ) {
17
18      Integer[] output = Arrays.stream(arr)
19          .boxed()
20          .toArray(Integer[]::new);
21
22
23      //      Regarding i -> Integer.bitCount(i) * 10000 + i to answer any potential
24      //      hashing the original numbers into another number generated from the count of
25      //      bits and then sorting the newly
26      //      generated numbers. so why 10000? simply because of the input range is 0 <=
27      //      arr[i] <= 10^4.
28      //      For instance [0,1,2,3,5,7], becomes something like this [0, 10001, 10002,
29      //      20003, 20005, 30007].
30
31      Arrays.sort(output, Comparator.comparingInt(value -> countSetBits(value) * 10000
32      + value));
33
34      int length = arr.length;
35
36      for (int i = 0; i < length; i++)
37          arr[i] = output[i];
38
39      return arr;
40  }
```

Your previous code was restored from your local storage. [Reset to default](#)

Testcase

Run Code Result

Debugger

Accepted

Runtime: 5 ms

Your input

[0,1,2,3,4,5,6,7,8]

Output

[0,1,2,4,8,3,5,6,7]

Diff

Expected

[0,1,2,4,8,3,5,6,7]

Console

Use Example Testcases

Run Code

Submit

Problems

Pick One

< Prev

1/1

Next >