Problem 1356: Sort Integers by The Number of 1 Bits

Problem Information

Difficulty: Easy
Acceptance Rate: 78.87%
Paid Only: No
Tags: Array, Bit Manipulation, Sorting, Counting

Problem Description

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: arr = [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`

Code Snippets

C++:

```
class Solution {
public:
vector<int> sortByBits(vector<int>& arr) {

}
};
```

Java:

```
class Solution {
public int[] sortByBits(int[] arr) {

}
}
```

Python3:

```
class Solution:
def sortByBits(self, arr: List[int]) -> List[int]:
```