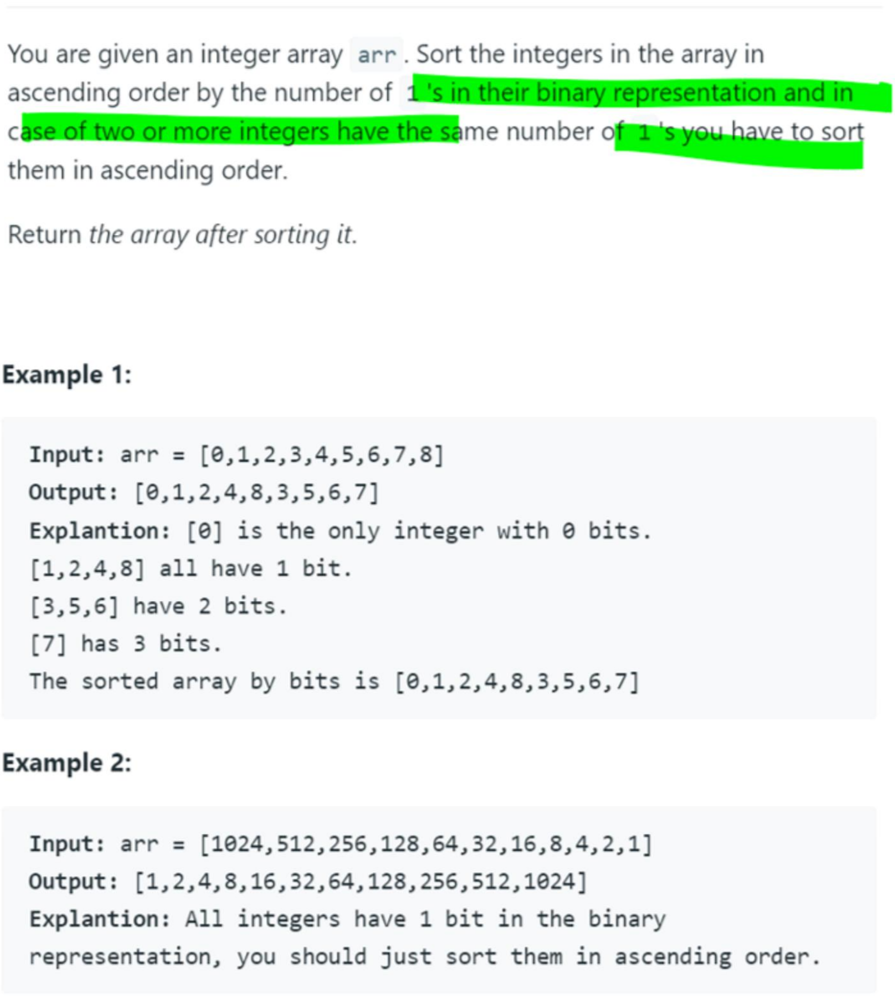
1356. Sort Integers by The Number of 1 Bits

28 June 2022 10:19 AM



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.

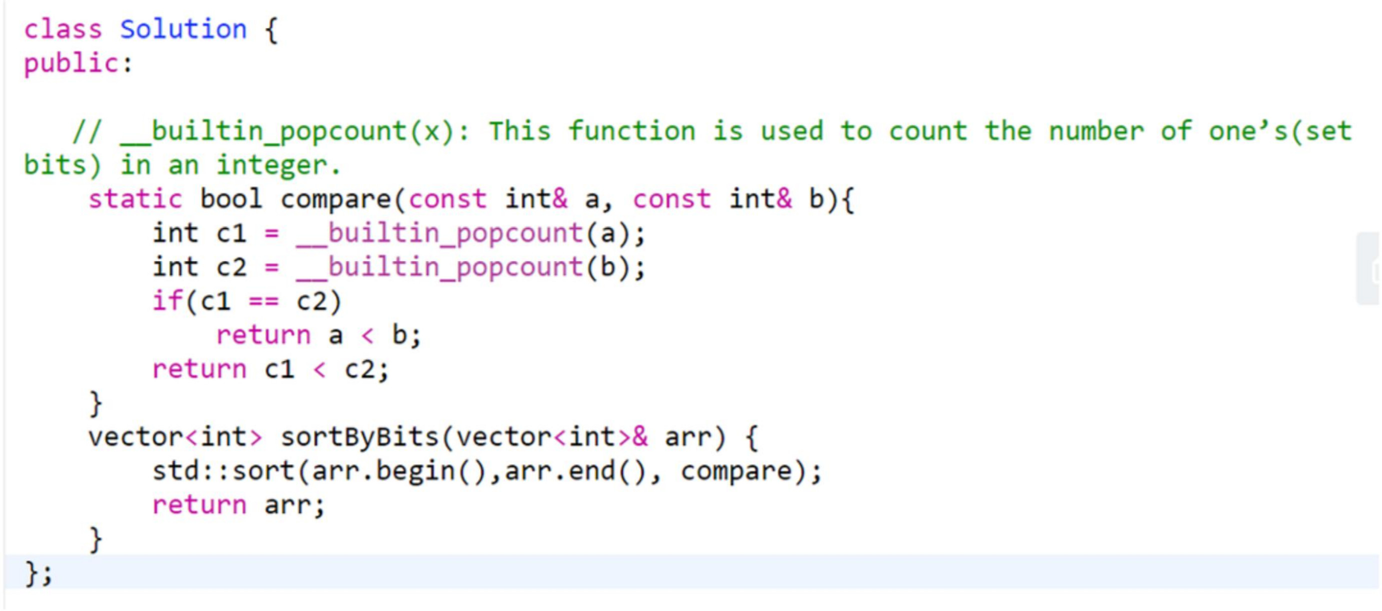
Return *the array after sorting it.*

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.
```



```
class Solution {
public:

   // __builtin_popcount(x): This function is used to count the number of one’s(set
bits) in an integer.
    static bool compare(const int& a, const int& b){
        int c1 = __builtin_popcount(a);
        int c2 = __builtin_popcount(b);
        if(c1 == c2)
            return a < b;
        return c1 < c2;
    }
    vector<int> sortByBits(vector<int>& arr) {
        std::sort(arr.begin(),arr.end(), compare);
        return arr;
    }
};
```