338. Counting Bits

Easy 5546 275 Add to List Share

Given an integer `n`, return *an array* `ans` *of length* `n + 1` *such that for each* `i` (`0 <= i <= n`), `ans[i]` *is the* ***number of*** `1` ***'s*** *in the binary representation of* `i`.

Example 1:

```
Input: n = 2
Output: [0,1,1]
Explanation:
0 --> 0
1 --> 1
2 --> 10
```

Example 2:

```
Input: n = 5
Output: [0,1,1,2,1,2]
Explanation:
0 --> 0
1 --> 1
2 --> 10
3 --> 11
4 --> 100
5 --> 101
```

Constraints:

- `0 <= n <= 10^5`

Follow up:

- It is very easy to come up with a solution with a runtime of `O(n log n)`. Can you do it in linear time `O(n)` and possibly in a single pass?
- Can you do it without using any built-in function (i.e., like `__builtin_popcount` in C++)?

Accepted 476,560 Submissions 656,905

Seen this question in a real interview before? Yes No

Companies

Related Topics

Similar Questions

Java Autocomplete

```java
class Solution {
    public int[] countBits(int n) {
        int[] bitCounts = new int[n + 1];

        for(int i = 0; i <= n; i++){
            int count = 0;
            int j = i;

            while(j != 0){
                count++;
                j = j & (j - 1);
            }

            bitCounts[i] = count;
        }

        return bitCounts;
    }
}
```

Your previous code was restored from your local storage. Reset to default