↩(/) Explore(/explore/)   Problems(/problemset/all/)   Interview   Contest   Discuss(/discuss/)   Store   ☆ Premium (/subscribe?ref=nb_npl)   🔔   ◊ 0   (/problems/deepest-leaves-sum/)   👤

6065. Largest Combination With Bitwise AND Greater Than Zero

My Submissions (/contest/weekly-contest-293/problems/largest-combination-with-bitwise-and-greater-than-zero/submissions/)

Back to Contest (/contest/weekly-contest-293/)

The **bitwise AND** of an array `nums` is the bitwise AND of all integers in `nums`.

- For example, for `nums = [1, 5, 3]`, the bitwise AND is equal to `1 & 5 & 3 = 1`.
- Also, for `nums = [7]`, the bitwise AND is `7`.

You are given an array of positive integers `candidates`. Evaluate the **bitwise AND** of every **combination** of numbers of `candidates`. Each number in `candidates` may only be used **once** in each combination.

Return *the size of the **largest** combination of* `candidates` *with a bitwise AND* **greater** *than* `0`.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

Example 1:

```
Input: candidates = [16,17,71,62,12,24,14]
Output: 4
Explanation: The combination [16,17,62,24] has a bitwise AND of 16 & 17 & 62 & 24 = 16 > 0.
The size of the combination is 4.
It can be shown that no combination with a size greater than 4 has a bitwise AND greater than 0.
Note that more than one combination may have the largest size.
For example, the combination [62,12,24,14] has a bitwise AND of 62 & 12 & 24 & 14 = 8 > 0.
```

Example 2:

```
Input: candidates = [8,8]
Output: 2
Explanation: The largest combination [8,8] has a bitwise AND of 8 & 8 = 8 > 0.
The size of the combination is 2, so we return 2.
```

Constraints:

- $1 <= candidates.length <= 10^5$
- $1 <= candidates[i] <= 10^7$

JavaScript ▾   ⟨⟩ ↻ ⚙

```javascript
1   const largestCombination = (a) => longestSubsequencePositiveBitWiseAND(a);
2
3   const longestSubsequencePositiveBitWiseAND = (a) => {
4       let bit = Array(32).fill(0);
5       for (let i = 0; i < a.length; i++) {
6           let pos = 31;
7           while (a[i] > 0) {
8               if (a[i] & 1) {
9                   bit[pos]++;
10              }
11              a[i] >>= 1;
12              pos--;
13          }
14      }
15      return Math.max(...bit);
16  };
```

Use Example Testcases

☐ **Custom Testcase**   Use Example Testcases

⏵ Run   ☁ Submit

Submission Result: Accepted (/submissions/detail/699718706/) ❓   More Details ❯ (/submissions/detail/699718706/)

Share your acceptance!

◀ 1

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

🇺🇸   United States (/region)