

1h 0m left

2. Bits in Archie

ALL

Alan Emtage is a computer scientist from Barbados who is widely known for implementing the world's first internet search engine, Archie. While implementing Archie, Emtage needs to test some bitwise functions required to make the search engine compute more efficiently.

Emtage takes an array of positive integers and a threshold value. The goal is to count the number of unordered pairs of elements in the array such that the sum of the number of set bits in the binary representation of their bitwise OR and bitwise AND is greater than or equal to the given threshold value.

**Example**

Consider Emtage has the array,  $arr = [2, 4, 6, 8, 10]$ , and the threshold value  $k = 4$ .

Let  $|$  denote the OR operation, and  $\&$  denote the AND operation, and  $/x/$  denotes the number of set bits in the binary representation of integer  $x$ .

There are 10 unordered pairs in Emtage's array:

- For pair  $(2, 4)$ ,  $|2|4| + |2\&4| = 2$ , which is less than threshold 4.
- For pair  $(2, 6)$ ,  $|2|6| + |2\&6| = 3$ , which is less than threshold 4.
- For pair  $(2, 8)$ ,  $|2|8| + |2\&8| = 2$ , which is less than threshold 4.
- For pair  $(2, 10)$ ,  $|2|10| + |2\&10| = 3$ , which is less than threshold 4.

Language C++20

```
1 > #include <bits/stdc++.h>
2
3 /*
4  * Complete the function
5  *
6  * The function is expected to return an integer.
7  * The function accepts two parameters:
8  * 1. INTEGER_ARRAY arr
9  * 2. INTEGER k
10 */
11
12 unordered_map<int, int> count_set_bits;
13 int countSetBits(int n) {
14     if(count_set_bits.find(n) != count_set_bits.end()) {
15         return count_set_bits[n];
16     }
17
18     int count = 0;
19     int temp = n;
20
21     while(temp > 0) {
22         if(temp & 1) {
23             count++;
24         }
25         temp >>= 1;
26     }
27
28     count_set_bits[n] = count;
29     return count;
30 }
31
32 long countBit(vector<int> arr, int k) {
33     int cnt = 0;
34     int n = arr.size();
35
36     for(int i = 0; i < n; i++) {
37         for(int j = i + 1; j < n; j++) {
38             int or_val = arr[i] | arr[j];
39             int and_val = arr[i] & arr[j];
40             int sum_bits = /or_val/ + /and_val/;
41             if(sum_bits >= k) {
42                 cnt++;
43             }
44         }
45     }
46     return cnt;
47 }
```

```
16
17 * 1. INTEGER_ARRAY arr
18 * 2. INTEGER k
19 */
20
21 unordered_map<int, int> count_set_bits;
22 int countSetBits(int n) {
23     if(count_set_bits.find(n) != count_set_bits.end()) {
24         return count_set_bits[n];
25     }
26
27     int count = 0;
28     int temp = n;
29
30     while(temp > 0) {
31         if(temp & 1) {
32             count++;
33         }
34         temp >>= 1;
35     }
36
37     count_set_bits[n] = count;
38     return count;
39 }
40
41 long countBit(vector<int> arr, int k) {
42     int cnt = 0;
43     int n = arr.size();
44
45     for(int i = 0; i < n; i++) {
46         for(int j = i + 1; j < n; j++) {
47             int or_val = arr[i] | arr[j];
48             int and_val = arr[i] & arr[j];
49             int sum_bits = /or_val/ + /and_val/;
50             if(sum_bits >= k) {
51                 cnt++;
52             }
53         }
54     }
55     return cnt;
56 }
```

```

34     temp >>= 1;
35 }
36     count_set_bits[n] = count;
37     return count;
38 }
39
40 long countBit(vector<int> arr, int k) {
41     int cnt = 0;
42     int n = arr.size();
43
44     for(int i = 0; i<n; i++){
45         for(int j = i + 1; j<n; j++) {
46             int orr = arr[i] | arr[j];
47             int andd = arr[i] & arr[j];
48             int tot_set_bits = countSetBits(orr) + countSetBits(andd);
49
50             if(tot_set_bits >= k) {
51                 cnt++;
52             }
53         }
54     }
55     return cnt;
56 }
57
58 > int main() ...

```

Test Results

Custom Input