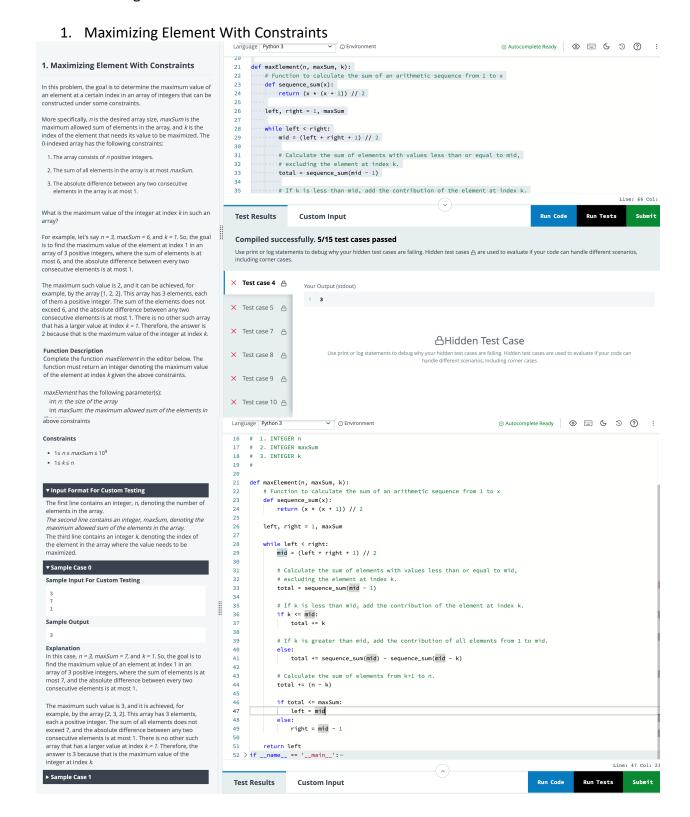
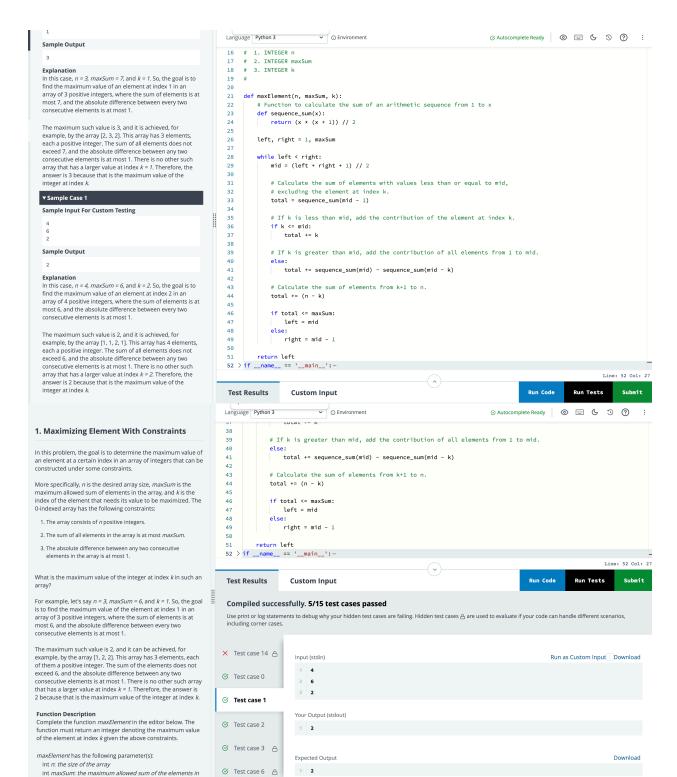
Problem Solving Intermediate





Bitwise AND ⊗ Autocomplete Ready | ⊚ ७ ৩ ? : 1 > #!/bin/python3--2. Bitwise AND # Complete the 'countPairs' function below. Given an array of non-negative integers, count the number of unordered pairs of array elements such that their <u>bitwise AND</u> # The function is expected to return a LONG INTEGER. 13 # The function accepts INTEGER_ARRAY arr as parameter. 15 For example, let's say the array is arr = [10, 7, 2, 8, 3], and let 16 17 from collections import defaultdict '&' denote the bitwise AND operator. There are 6 unordered pairs of its elements that have a bitwise AND that is a power of two: def countPairs(arr): # Write your code here
po2 = lambda x: x > 0 and not(x & (x - 1)) 19 • For indices (0,1), 10 & 7 = 2, which is a power of 2. 20 21 d = defaultdict(int) for x in arr: • For indices (0,2), 10 & 2 = 2, which is a power of 2. d[x] += 1 For indices (0.3), 10 & 8 = 8, which is a power of 2. d = list(d.items()) 23 For indices (0,4), 10 & 3 = 2, which is a power of 2. 24 25 • For indices (1,2), 7 & 2 = 2, which is a power of 2. for i in range(len(d)): • For indices (2,4), 2 & 3 = 2, which is a power of 2. a, a_cnt = d[i]
for j in range(i, len(d)):
b, b_cnt = d[j] 26 27 28 Therefore, the answer is 6. 29 if po2(a & b): **Function Description** 30 if a ==b: ans += (a_cnt * (a_cnt - 1)) // 2 Complete the function countPairs in the editor below. else: countPairs has the following parameter: ans += a_cnt * b_cnt int arr[n]: an array of integers 34 return ans Returns: 36 > if __name__ == '__main__': -int: the number of unordered pairs of elements of *arr* such that their bitwise AND is a power of 2 • $0 \le arr[i] < 2^{12}$ Line: 10 Col: 1 The first line contains an integer, n, denoting the number of Test Results **Custom Input** Run Code Run Tests Submit Each line i of the n subsequent lines (where $0 \le i < n$) • ① Environment ▼ Input Format For Custom Testing Language Python 3 The first line contains an integer, *n*, denoting the number of elements in *arr*. d = defaultdict(int) \odot for x in arr:
 d[x] += 1 21 Each line i of the n subsequent lines (where $0 \le i < n$) 22 contains an integer describing arr[i]. 23 d = list(d.items()) ALL ▼ Sample Case 0 25 26 for i in range(len(d)): Sample Input For Custom Testing (1) a, a_cnt = d[i] for j in range(i, len(d)):
 b, b_cnt = d[j]
 if po2(a & b): STDIN Function 27 28 => n = 4 => arr = [1, 2, 1, 3] 29 if a ==b: ans += (a_cnt * (a_cnt - 1)) // 2 30 31 32 else: ans += a_cnt * b_cnt Sample Output 34 return ans Run Code Submit All unordered pair of elements whose bitwise AND is a Test Results Custom Input Run Tests power of 2 are: Compiled successfully. All available test cases passed • For indices (0,2), 1 & 1 = 1, which is a power of 2. • For indices (0,3), 1 & 3 = 1, which is a power of 2. For indices (1,3), 2 & 3 = 2, which is a power of 2. For indices (2,3), 1 & 3 = 1, which is a power of 2. Run as Custom Input | Download Input (stdin) 1 4 ▼ Sample Case 1 3 3 Sample Input For Custom Testing ⊘ Test case 3 🛆 Your Output (stdout) Ø Test case 4 △ Sample Output There are no pairs of array elements such that their bitwise Expected Output AND is a power of 2. Therefore, the answer is 0. ⊘ Test case 6 💍