Problem 2172: Maximum AND Sum of Array

Problem Information

Difficulty: Hard
Acceptance Rate: 50.41%
Paid Only: No
Tags: Array, Dynamic Programming, Bit Manipulation, Bitmask

Problem Description

You are given an integer array `nums` of length `n` and an integer `numSlots` such that `2 * numSlots >= n`. There are `numSlots` slots numbered from `1` to `numSlots`.

You have to place all `n` integers into the slots such that each slot contains at **most** two numbers. The **AND sum** of a given placement is the sum of the **bitwise** `AND` of every number with its respective slot number.

* For example, the **AND sum** of placing the numbers `[1, 3]` into slot _`1`_ and `[4, 6]` into slot _`2`_ is equal to `(1 AND _1_) + (3 AND _1_) + (4 AND _2_) + (6 AND _2_) = 1 + 1 + 0 + 2 = 4`.

Return _the maximum possible**AND sum** of _`nums` _given_`numSlots` _slots._

Example 1:

Input: nums = [1,2,3,4,5,6], numSlots = 3 **Output:** 9 **Explanation:** One possible placement is [1, 4] into slot _1_ , [2, 6] into slot _2_ , and [3, 5] into slot _3_. This gives the maximum AND sum of (1 AND _1_) + (4 AND _1_) + (2 AND _2_) + (6 AND _2_) + (3 AND _3_) + (5 AND _3_) = 1 + 0 + 2 + 2 + 3 + 1 = 9.

Example 2:

Input: nums = [1,3,10,4,7,1], numSlots = 9 **Output:** 24 **Explanation:** One possible placement is [1, 1] into slot _1_ , [3] into slot _3_ , [4] into slot _4_ , [7] into slot _7_ , and [10] into slot _9_. This gives the maximum AND sum of (1 AND _1_) + (1 AND _1_) + (3 AND _3_) + (4 AND _4_) + (7 AND _7_) + (10 AND _9_) = 1 + 1 + 3 + 4 + 7 + 8 = 24. Note that slots 2,

5, 6, and 8 are empty which is permitted.

Constraints:

* `n == nums.length` * `1 <= numSlots <= 9` * `1 <= n <= 2 * numSlots` * `1 <= nums[i] <= 15`

Code Snippets

C++:

```
class Solution {
public:
int maximumANDSum(vector<int>& nums, int numSlots) {

}
};
```

Java:

```
class Solution {
public int maximumANDSum(int[] nums, int numSlots) {

}
}
```

Python3:

```
class Solution:
def maximumANDSum(self, nums: List[int], numSlots: int) -> int:
```