Problem 1799: Maximize Score After N Operations

Problem Information

Difficulty: Hard
Acceptance Rate: 57.95%
Paid Only: No
Tags: Array, Math, Dynamic Programming, Backtracking, Bit Manipulation, Number Theory, Bitmask

Problem Description

You are given `nums`, an array of positive integers of size `2 * n`. You must perform `n` operations on this array.

In the `ith` operation **(1-indexed)** , you will:

* Choose two elements, `x` and `y`. * Receive a score of `i * gcd(x, y)`. * Remove `x` and `y` from `nums`.

Return _the maximum score you can receive after performing_`n` _operations._

The function `gcd(x, y)` is the greatest common divisor of `x` and `y`.

Example 1:

Input: nums = [1,2] **Output:** 1 **Explanation:** The optimal choice of operations is: (1 * gcd(1, 2)) = 1

Example 2:

Input: nums = [3,4,6,8] **Output:** 11 **Explanation:** The optimal choice of operations is: (1 * gcd(3, 6)) + (2 * gcd(4, 8)) = 3 + 8 = 11

Example 3:

Input: nums = [1,2,3,4,5,6] **Output:** 14 **Explanation:** The optimal choice of operations is: (1 * gcd(1, 5)) + (2 * gcd(2, 4)) + (3 * gcd(3, 6)) = 1 + 4 + 9 = 14

Constraints:

* `1 <= n <= 7` * `nums.length == 2 * n` * `1 <= nums[i] <= 106`

Code Snippets

C++:

```
class Solution {
public:
int maxScore(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public int maxScore(int[] nums) {

}
}
```

Python3:

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