

# Problem 2862: Maximum Element-Sum of a Complete Subset of Indices

## Problem Information

**Difficulty:** Hard

**Acceptance Rate:** 42.32%

**Paid Only:** No

**Tags:** Array, Math, Number Theory

## Problem Description

You are given a **1-indexed** array `nums`. Your task is to select a **complete subset** from `nums` where every pair of selected indices multiplied is a perfect square,. i. e. if you select `ai` and `aj`,  $i * j$  must be a perfect square.

Return the `_sum_` of the complete subset with the `_maximum sum_`.

**Example 1:**

**Input:** nums = [8,7,3,5,7,2,4,9]

**Output:** 16

**Explanation:**

We select elements at indices 2 and 8 and `2 \* 8` is a perfect square.

**Example 2:**

**Input:** nums = [8,10,3,8,1,13,7,9,4]

**Output:** 20

**Explanation:**

We select elements at indices 1, 4, and 9. `1 \* 4`, `1 \* 9`, `4 \* 9` are perfect squares.

**\*\*Constraints:\*\***

\* `1 <= n == nums.length <= 104` \* `1 <= nums[i] <= 109`

## Code Snippets

**C++:**

```
class Solution {
public:
    long long maximumSum(vector<int>& nums) {
        }
};
```

**Java:**

```
class Solution {
public long maximumSum(List<Integer> nums) {
        }
}
```

**Python3:**

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