Problem 3186: Maximum Total Damage With Spell Casting

Problem Information

Difficulty: Medium
Acceptance Rate: 45.02%
Paid Only: No
Tags: Array, Hash Table, Two Pointers, Binary Search, Dynamic Programming, Sorting, Counting

Problem Description

A magician has various spells.

You are given an array `power`, where each element represents the damage of a spell. Multiple spells can have the same damage value.

It is a known fact that if a magician decides to cast a spell with a damage of `power[i]`, they **cannot** cast any spell with a damage of `power[i] - 2`, `power[i] - 1`, `power[i] + 1`, or `power[i] + 2`.

Each spell can be cast **only once**.

Return the **maximum** possible _total damage_ that a magician can cast.

Example 1:

Input: power = [1,1,3,4]

Output: 6

Explanation:

The maximum possible damage of 6 is produced by casting spells 0, 1, 3 with damage 1, 1, 4.

Example 2:

Input: power = [7,1,6,6]

Output: 13

Explanation:

The maximum possible damage of 13 is produced by casting spells 1, 2, 3 with damage 1, 6, 6.

Constraints:

* `1 <= power.length <= 105` * `1 <= power[i] <= 109`

Code Snippets

C++:

```
class Solution {
public:
long long maximumTotalDamage(vector<int>& power) {

}
};
```

Java:

```
class Solution {
public long maximumTotalDamage(int[] power) {

}
}
```

Python3:

```
class Solution:
def maximumTotalDamage(self, power: List[int]) -> int:
```