

Problem 3147: Taking Maximum Energy From the Mystic Dungeon

Problem Information

Difficulty: Medium

Acceptance Rate: 60.92%

Paid Only: No

Tags: Array, Prefix Sum

Problem Description

In a mystic dungeon, n magicians are standing in a line. Each magician has an attribute that gives you energy. Some magicians can give you negative energy, which means taking energy from you.

You have been cursed in such a way that after absorbing energy from magician i , you will be instantly transported to magician $(i + k)$. This process will be repeated until you reach the magician where $(i + k)$ does not exist.

In other words, you will choose a starting point and then teleport with k jumps until you reach the end of the magicians' sequence, **absorbing all the energy** during the journey.

You are given an array `energy` and an integer `k`. Return the **maximum** possible energy you can gain.

Note that when you reach a magician, you must take energy from them, whether it is negative or positive energy.

Example 1:

Input: `energy = [5,2,-10,-5,1]`, `k = 3`

Output: 3

Explanation: We can gain a total energy of 3 by starting from magician 1 absorbing $2 + 1 = 3$.

****Example 2:****

****Input:**** energy = [-2,-3,-1], k = 2

****Output:**** -1

****Explanation:**** We can gain a total energy of -1 by starting from magician 2.

****Constraints:****

$1 \leq \text{energy.length} \leq 105$ $-1000 \leq \text{energy}[i] \leq 1000$ $1 \leq k \leq \text{energy.length} - 1$

Code Snippets

C++:

```
class Solution {
public:
    int maximumEnergy(vector<int>& energy, int k) {

    }
};
```

Java:

```
class Solution {
    public int maximumEnergy(int[] energy, int k) {

    }
}
```

Python3:

```
class Solution:
    def maximumEnergy(self, energy: List[int], k: int) -> int:
```