

Problem 2517: Maximum Tastiness of Candy Basket

Problem Information

Difficulty: Medium

Acceptance Rate: 67.47%

Paid Only: No

Tags: Array, Binary Search, Greedy, Sorting

Problem Description

You are given an array of positive integers `price` where `price[i]` denotes the price of the `ith` candy and a positive integer `k`.

The store sells baskets of `k` **distinct** candies. The **tastiness** of a candy basket is the smallest absolute difference of the **prices** of any two candies in the basket.

Return _the**maximum** tastiness of a candy basket._

Example 1:

Input: price = [13,5,1,8,21,2], k = 3 **Output:** 8 **Explanation:** Choose the candies with the prices [13,5,21]. The tastiness of the candy basket is: $\min(|13 - 5|, |13 - 21|, |5 - 21|) = \min(8, 8, 16) = 8$. It can be proven that 8 is the maximum tastiness that can be achieved.

Example 2:

Input: price = [1,3,1], k = 2 **Output:** 2 **Explanation:** Choose the candies with the prices [1,3]. The tastiness of the candy basket is: $\min(|1 - 3|) = \min(2) = 2$. It can be proven that 2 is the maximum tastiness that can be achieved.

Example 3:

Input: price = [7,7,7,7], k = 2 **Output:** 0 **Explanation:** Choosing any two distinct candies from the candies we have will result in a tastiness of 0.

****Constraints:****

* `2 <= k <= price.length <= 105` * `1 <= price[i] <= 109`

Code Snippets

C++:

```
class Solution {  
public:  
    int maximumTastiness(vector<int>& price, int k) {  
  
    }  
};
```

Java:

```
class Solution {  
public int maximumTastiness(int[] price, int k) {  
  
}  
}
```

Python3:

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