Problem 3075: Maximize Happiness of Selected Children

Problem Information

Difficulty: Medium
Acceptance Rate: 54.91%
Paid Only: No
Tags: Array, Greedy, Sorting

Problem Description

You are given an array `happiness` of length `n`, and a **positive** integer `k`.

There are `n` children standing in a queue, where the `ith` child has **happiness value** `happiness[i]`. You want to select `k` children from these `n` children in `k` turns.

In each turn, when you select a child, the **happiness value** of all the children that have **not** been selected till now decreases by `1`. Note that the happiness value **cannot** become negative and gets decremented **only** if it is positive.

Return _the**maximum** sum of the happiness values of the selected children you can achieve by selecting _`k` _children_.

Example 1:

Input: happiness = [1,2,3], k = 2 **Output:** 4 **Explanation:** We can pick 2 children in the following way: - Pick the child with the happiness value == 3. The happiness value of the remaining children becomes [0,1]. - Pick the child with the happiness value == 1. The happiness value of the remaining child becomes [0]. Note that the happiness value cannot become less than 0. The sum of the happiness values of the selected children is 3 + 1 = 4.

Example 2:

Input: happiness = [1,1,1,1], k = 2 **Output:** 1 **Explanation:** We can pick 2 children in the following way: - Pick any child with the happiness value == 1. The happiness value of the remaining children becomes [0,0,0]. - Pick the child with the happiness value == 0. The happiness value of the remaining child becomes [0,0]. The sum of the happiness values of the selected children is 1 + 0 = 1.

Example 3:

Input: happiness = [2,3,4,5], k = 1 **Output:** 5 **Explanation:** We can pick 1 child in the following way: - Pick the child with the happiness value == 5. The happiness value of the remaining children becomes [1,2,3]. The sum of the happiness values of the selected children is 5.

Constraints:

* `1 <= n == happiness.length <= 2 * 105` * `1 <= happiness[i] <= 108` * `1 <= k <= n`

Code Snippets

C++:

```
class Solution {
public:
long long maximumHappinessSum(vector<int>& happiness, int k) {

}
};
```

Java:

```
class Solution {
public long maximumHappinessSum(int[] happiness, int k) {

}
}
```

Python3:

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