

Problem 2813: Maximum Elegance of a K-Length Subsequence

Problem Information

Difficulty: Hard

Acceptance Rate: 28.32%

Paid Only: No

Tags: Array, Hash Table, Stack, Greedy, Sorting, Heap (Priority Queue)

Problem Description

You are given a **0-indexed** 2D integer array `items` of length `n` and an integer `k`.

`items[i] = [profiti, categoryi]`, where `profiti` and `categoryi` denote the profit and category of the `i`th item respectively.

Let's define the **elegance** of a **subsequence** of `items` as `total_profit + distinct_categories2`, where `total_profit` is the sum of all profits in the subsequence, and `distinct_categories` is the number of **distinct** categories from all the categories in the selected subsequence.

Your task is to find the **maximum elegance** from all subsequences of size `k` in `items`.

Return `_` an integer denoting the maximum elegance of a subsequence of `items` with size exactly `k`.

Note: A subsequence of an array is a new array generated from the original array by deleting some elements (possibly none) without changing the remaining elements' relative order.

Example 1:

Input: `items = [[3,2],[5,1],[10,1]]`, `k = 2` **Output:** 17 **Explanation:** In this example, we have to select a subsequence of size 2. We can select `items[0] = [3,2]` and `items[2] = [10,1]`. The total profit in this subsequence is `3 + 10 = 13`, and the subsequence contains 2 distinct categories `[2,1]`. Hence, the elegance is `13 + 22 = 17`, and we can show that it is the

maximum achievable elegance.

Example 2:

Input: items = [[3,1],[3,1],[2,2],[5,3]], k = 3 **Output:** 19 **Explanation:** In this example, we have to select a subsequence of size 3. We can select items[0] = [3,1], items[2] = [2,2], and items[3] = [5,3]. The total profit in this subsequence is $3 + 2 + 5 = 10$, and the subsequence contains 3 distinct categories [1,2,3]. Hence, the elegance is $10 + 3 \times 2 = 19$, and we can show that it is the maximum achievable elegance.

Example 3:

Input: items = [[1,1],[2,1],[3,1]], k = 3 **Output:** 7 **Explanation:** In this example, we have to select a subsequence of size 3. We should select all the items. The total profit will be $1 + 2 + 3 = 6$, and the subsequence contains 1 distinct category [1]. Hence, the maximum elegance is $6 + 1 \times 1 = 7$.

Constraints:

$1 \leq \text{items.length} \leq n \leq 10^5$ $1 \leq \text{items}[i].\text{length} \leq 2$ $1 \leq \text{items}[i][0] \leq \text{profit}_i$ $1 \leq \text{items}[i][1] \leq \text{category}_i$ $1 \leq \text{profit}_i \leq 10^9$ $1 \leq \text{category}_i \leq n$ $1 \leq k \leq n$

Code Snippets

C++:

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

    }
};
```

Java:

```
class Solution {
    public long findMaximumElegance(int[][] items, int k) {

    }
}
```

Python3:

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