

Problem 2611: Mice and Cheese

Problem Information

Difficulty: Medium

Acceptance Rate: 47.96%

Paid Only: No

Tags: Array, Greedy, Sorting, Heap (Priority Queue)

Problem Description

There are two mice and `n` different types of cheese, each type of cheese should be eaten by exactly one mouse.

A point of the cheese with index `i` (**0-indexed**) is:

* `reward1[i]` if the first mouse eats it.
* `reward2[i]` if the second mouse eats it.

You are given a positive integer array `reward1`, a positive integer array `reward2`, and a non-negative integer `k`.

Return **the maximum** points the mice can achieve if the first mouse eats exactly `k` types of cheese.

Example 1:

Input: reward1 = [1,1,3,4], reward2 = [4,4,1,1], k = 2
Output: 15
Explanation: In this example, the first mouse eats the 2nd (0-indexed) and the 3rd types of cheese, and the second mouse eats the 0th and the 1st types of cheese. The total points are $4 + 4 + 3 + 4 = 15$. It can be proven that 15 is the maximum total points that the mice can achieve.

Example 2:

Input: reward1 = [1,1], reward2 = [1,1], k = 2
Output: 2
Explanation: In this example, the first mouse eats the 0th (0-indexed) and 1st types of cheese, and the second mouse does not eat any cheese. The total points are $1 + 1 = 2$. It can be proven that 2 is the maximum total points that the mice can achieve.

****Constraints:****

```
* `1 <= n == reward1.length == reward2.length <= 105` * `1 <= reward1[i], reward2[i] <= 1000`  
* `0 <= k <= n`
```

Code Snippets

C++:

```
class Solution {  
public:  
    int miceAndCheese(vector<int>& reward1, vector<int>& reward2, int k) {  
  
    }  
};
```

Java:

```
class Solution {  
public int miceAndCheese(int[] reward1, int[] reward2, int k) {  
  
}  
}
```

Python3:

```
class Solution:  
    def miceAndCheese(self, reward1: List[int], reward2: List[int], k: int) ->  
        int:
```