Problem 3572: Maximize Y■Sum by Picking a Triplet of Distinct X■Values

Problem Information

Difficulty: Medium
Acceptance Rate: 62.85%
Paid Only: No
Tags: Array, Hash Table, Greedy, Sorting, Heap (Priority Queue)

Problem Description

You are given two integer arrays `x` and `y`, each of length `n`. You must choose three **distinct** indices `i`, `j`, and `k` such that:

* `x[i] != x[j]` * `x[j] != x[k]` * `x[k] != x[i]`

Your goal is to **maximize** the value of `y[i] + y[j] + y[k]` under these conditions. Return the **maximum** possible sum that can be obtained by choosing such a triplet of indices.

If no such triplet exists, return -1.

Example 1:

Input: x = [1,2,1,3,2], y = [5,3,4,6,2]

Output: 14

Explanation:

* Choose `i = 0` (`x[i] = 1`, `y[i] = 5`), `j = 1` (`x[j] = 2`, `y[j] = 3`), `k = 3` (`x[k] = 3`, `y[k] = 6`). * All three values chosen from `x` are distinct. `5 + 3 + 6 = 14` is the maximum we can obtain. Hence, the output is 14.

Example 2:

Input: x = [1,2,1,2], y = [4,5,6,7]

Output: -1

Explanation:

* There are only two distinct values in `x`. Hence, the output is -1.

Constraints:

* `n == x.length == y.length` * `3 <= n <= 105` * `1 <= x[i], y[i] <= 106`

Code Snippets

C++:

```
class Solution {
public:
int maxSumDistinctTriplet(vector<int>& x, vector<int>& y) {

}
};
```

Java:

```
class Solution {
public int maxSumDistinctTriplet(int[] x, int[] y) {

}
}
```

Python3:

```
class Solution:
def maxSumDistinctTriplet(self, x: List[int], y: List[int]) -> int:
```