Problem 2600: K Items With the Maximum Sum

Problem Information

Difficulty: Easy
Acceptance Rate: 60.04%
Paid Only: No
Tags: Math, Greedy

Problem Description

There is a bag that consists of items, each item has a number `1`, `0`, or `-1` written on it.

You are given four **non-negative** integers `numOnes`, `numZeros`, `numNegOnes`, and `k`.

The bag initially contains:

* `numOnes` items with `1`s written on them. * `numZeroes` items with `0`s written on them. * `numNegOnes` items with `-1`s written on them.

We want to pick exactly `k` items among the available items. Return _the**maximum** possible sum of numbers written on the items_.

Example 1:

Input: numOnes = 3, numZeros = 2, numNegOnes = 0, k = 2 **Output:** 2 **Explanation:** We have a bag of items with numbers written on them {1, 1, 1, 0, 0}. We take 2 items with 1 written on them and get a sum in a total of 2. It can be proven that 2 is the maximum possible sum.

Example 2:

Input: numOnes = 3, numZeros = 2, numNegOnes = 0, k = 4 **Output:** 3 **Explanation:** We have a bag of items with numbers written on them {1, 1, 1, 0, 0}. We take 3 items with 1 written on them, and 1 item with 0 written on it, and get a sum in a total of 3. It can be proven that 3 is the maximum possible sum.

Constraints:

* `0 <= numOnes, numZeros, numNegOnes <= 50` * `0 <= k <= numOnes + numZeros + numNegOnes`

Code Snippets

C++:

```
class Solution {
public:
int kItemsWithMaximumSum(int numOnes, int numZeros, int numNegOnes, int k) {

}
};
```

Java:

```
class Solution {
public int kItemsWithMaximumSum(int numOnes, int numZeros, int numNegOnes,
int k) {

}
}
```

Python3:

```
class Solution:
def kItemsWithMaximumSum(self, numOnes: int, numZeros: int, numNegOnes: int,
k: int) -> int:
```