



Java Greedy

Assignment Questions

Q. 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 1s written on them.

numZeros items with 0s written on them.

numNegOnes items with -1s 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.

Q. Given an integer k, return the minimum number of Fibonacci numbers whose sum is equal to k. The same Fibonacci number can be used multiple times.

The Fibonacci numbers are defined as:

$F_1 = 1$

$F_2 = 1$

$F_n = F_{n-1} + F_{n-2}$ for $n > 2$.

It is guaranteed that for the given constraints we can always find such Fibonacci numbers that sum up to k.

Example 1:

Input: k = 7

Output: 2

Explanation: The Fibonacci numbers are: 1, 1, 2, 3, 5, 8, 13, ...

For k = 7 we can use 2 + 5 = 7.

Example 2:

Input: $k = 10$

Output: 2

Explanation: For $k = 10$ we can use $2 + 8 = 10$.

Example 3:

Input: $k = 19$

Output: 3

Explanation: For $k = 19$ we can use $1 + 5 + 13 = 19$.

Q. You are given a non-negative integer array `nums`. In one operation, you must:

Choose a positive integer x such that x is less than or equal to the smallest non-zero element in `nums`.
Subtract x from every positive element in `nums`.
Return the minimum number of operations to make every element in `nums` equal to 0.

Example 1:

Input: `nums = [1,5,0,3,5]`

Output: 3

Explanation:

In the first operation, choose $x = 1$. Now, `nums = [0,4,0,2,4]`.

In the second operation, choose $x = 2$. Now, `nums = [0,2,0,0,2]`.

In the third operation, choose $x = 2$. Now, `nums = [0,0,0,0,0]`.

Example 2:

Input: `nums = [0]`

Output: 0

Explanation:

Each element in `nums` is already 0 so no operations are needed.

Q. You are given a positive integer n , you can do the following operation any number of times:

Add or subtract a power of 2 from n .

Return the minimum number of operations to make n equal to 0.

A number x is power of 2 if $x == 2^i$ where $i \geq 0$.

Example 1:

Input: $n = 39$

Output: 3

Explanation: We can do the following operations:

- Add $20 = 1$ to n , so now $n = 40$.
- Subtract $23 = 8$ from n , so now $n = 32$.
- Subtract $25 = 32$ from n , so now $n = 0$.

It can be shown that 3 is the minimum number of operations we need to make n equal to 0.

Example 2:

Input: $n = 54$

Output: 3

Explanation: We can do the following operations:

- Add $21 = 2$ to n , so now $n = 56$.
- Add $23 = 8$ to n , so now $n = 64$.
- Subtract $26 = 64$ from n , so now $n = 0$.

So the minimum number of operations is 3.

Q. Given two arrays $a[]$ and $b[]$ of the same size. Task is to find the minimum sum of two elements such that they belong to different arrays and are not at the same index in their arrays.

Example :

Input:

$a[] = \{5, 4, 13, 2, 1\}$

$b[] = \{2, 3, 4, 6, 5\}$

Output:

We take 1 from $a[]$ and 2 from $b[]$

Sum is $1 + 2 = 3$.

Input:

$a[] = \{5, 4, 13, 1\}$

$b[] = \{3, 2, 6, 1\}$

Output: 3

We take 1 from $a[]$ and 2 from $b[]$. Note that we can't take 1 from $b[]$ as the elements can not be at same index.



**THANK
YOU !**

