You have three stacks of cylinders where each cylinder has the same diameter, but they may vary in height. You can change the height of a stack by removing and discarding its topmost cylinder any number of times.

Find the maximum possible height of the stacks such that all of the stacks are exactly the same height. This means you must remove zero or more cylinders from the top of zero or more of the three stacks until they are all the same height, then return the height.

Example

$h1 = [1, 2, 1, 1]$
$h2 = [1, 1, 2]$
$h3 = [1, 1]$

There are $4, 3$ and 2 cylinders in the three stacks, with their heights in the three arrays. Remove the top 2 cylinders from $h1$ (heights = [1, 2]) and from $h2$ (heights = [1, 1]) so that the three stacks all are 2 units tall. Return 2 as the answer.

Note: An empty stack is still a stack.

Function Description

Complete the equalStacks function in the editor below.

equalStacks has the following parameters:

- int h1[n1]: the first array of heights
- int h2[n2]: the second array of heights
- int h3[n3]: the third array of heights

Returns

- int: the height of the stacks when they are equalized

Input Format

The first line contains three space-separated integers, $n1$, $n2$, and $n3$, the numbers of cylinders in stacks 1, 2, and 3. The subsequent lines describe the respective heights of each cylinder in a stack from top to bottom:

- The second line contains $n1$ space-separated integers, the cylinder heights in stack 1. The first element is the top cylinder of the stack.
- The third line contains $n2$ space-separated integers, the cylinder heights in stack 2. The first element is the top cylinder of the stack.
- The fourth line contains $n3$ space-separated integers, the cylinder heights in stack 3. The first element is the top cylinder of the stack.

Constraints

- $0 < n1, n2, n3 \leq 10^5$
- $0 < \ height\ of\ any\ cylinder\ \leq 100$

Change Theme Language Python 3

```python
# The function accepts following parameters:
#  1. INTEGER_ARRAY h1
#  2. INTEGER_ARRAY h2
#  3. INTEGER_ARRAY h3
#
def largest_common(a1, a2, a3):
    (hash2, hash3) = (set(a2), set(a3))
    common = []
    for ele in a1:
        if ele in hash2 and ele in hash3: common.append(ele)

    if len(common): return max(common)
    return 0

def equalStacks(h1, h2, h3):
    # Write your code here
    (len_h1, len_h2, len_h3) = (len(h1), len(h2), len(h3))

    sum1 = [0 for x in range(len_h1)]
    sum2 = [0 for x in range(len_h2)]
    sum3 = [0 for x in range(len_h3)]

    (sum1[len_h1-1], sum2[len_h2-1], sum3[len_h3-1]) =\
    (h1[len_h1-1], h2[len_h2-1], h3[len_h3-1])

    for idx in reversed(range(len_h1-1)):
        sum1[idx] = h1[idx] + sum1[idx+1]

    for idx in reversed(range(len_h2-1)):
        sum2[idx] = h2[idx] + sum2[idx+1]

    for idx in reversed(range(len_h3-1)):
        sum3[idx] = h3[idx] + sum3[idx+1]

    return largest_common(sum1, sum2, sum3)

```

Line: 85 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge