Algorithm Design Laboratory with Applications

Prof. Stefano Leucci

Problem: *A massive bookworm.*

The university library is replacing some books with new copies, and is gifting the old copies to students. The old copies are arranged in two piles S_1 e S_2 containing n and m books, respectively. You can take any number of books from the top of S_1, and any number of books from the top of S_2, but you cannot take a book from a pile without also taking all the books above it.

Each book has a certain weight in grams (a positive integer). Your backpack can hold up to $W \in \mathbb{N}^+$ grams, and your goal is that taking the largest number η of books from the two piles without exceeding the (overall) weight of W grams.

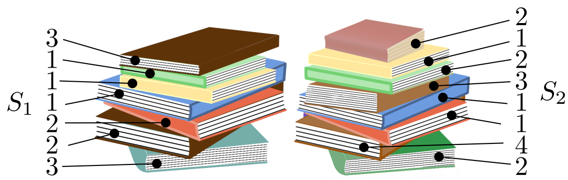
Design an algorithm that, given S_1, S_2, W, and the weight of each book, returns η.

Input. The input consists of a set of instances, or *test-cases*, of the previous problem. The first line of the input contains the number T of test-cases. The first line of each test-case contains the integers n, m, and W. The second line of each test-case contains n integers $w_1, \dots, w_n$, where w_i is the weight of the i-th book from the top of S_1. Finally, the third and last line of each test-case contains m integers $w'_1, \dots, w'_m$, where w'_i is the weight of the i-th book from the top of S_2.

Output. The output consists of T lines. The i-th lines is the solution to the i-th test case and contains η.

Assumptions. $1 \le T \le 10$; $1 \le n, m \le 2^{19}$; $W \le 2^{30}$.
Each book weighs at most 2^{11} grams.

Example. If $W = 9$, the weights of the books in S_1 are $\langle 3, 1, 1, 1, 2, 2, 3 \rangle$ (from the top to the bottom of the stack), and those of the books in S_2 are $\langle 2, 1, 2, 3, 1, 1, 4, 2 \rangle$, the optimal value of η is 6 and can be attained by taking 4 books from S_1 and 2 books from S_2.



Input:

```
1
7 8 9
3 1 1 1 2 2 3
2 1 2 3 1 1 4 2
```

Output:

```
6
```

Requirements. Your algorithm must have an asymptotic time complexity of $O(n + m)$.

Notes. A reasonable implementation should not require more than 1 second for each input file.