Problem Statement

As of late, your usually high-performance computer has been acting rather sluggish. You come to realize that while you have plenty of free disk space on your machine, it is split up over many hard drives. You decide that the secret to improving performance is to consolidate all the data on your computer onto as few hard drives as possible.

Given a int[] **used**, representing the amount of disk space used on each drive, and a corresponding int[] **total** , representing the total capacity of each drive mentioned in **used**, you should return the minimum number of hard drives needed to store the data after the consolidation is complete. You may assume that the data consists of very small files, such that splitting it up and moving parts of it onto different hard drives never presents a problem.

Definition

Class: DiskSpace

Method signature: public int minDrives(int[] used, int[] total)

Constraints

- **used** will contain between 1 and 50 elements, inclusive.
- **used** and **total** will contain the same number of elements.
- Each element of **used** will be between 1 and 1000, inclusive.
- Each element of **total** will be between 1 and 1000, inclusive.
- `used[i]` will always be less than or equal to `total[i]`, for every valid index `i`.

Examples

0)

```
{300,525,110}

{350,600,115}

Returns: 2
```

In this example, the computer contains three hard drives:

- Hard drive 1: 350 MB total, 300 MB used, 50 MB free
- Hard drive 2: 600 MB total, 525 MB used, 75 MB free
- Hard drive 3: 115 MB total, 110 MB used, 5 MB free

One way to pack the data onto as few drives as possible is as follows. First, move 50 MB from hard drive 3 to hard drive 1, completely filling it up. Next, move the remaining 60 MB from hard drive 3 to hard drive 2. There are still two hard drives which contain data after this process, so your method should return 2.

1)

```
{1,200,200,199,200,200}

{1000,200,200,200,200,200}

Returns: 1
```

One way to consolidate the data would be to move the 1 MB from hard drive 1 to hard drive 4. However, this is a poor choice, as it results in only one free hard drive and five hard drives which still contain data. A better decision is to move all the data from the other five hard drives onto hard drive 1. Now there is only one hard drive which contains data. Since this is the optimal strategy, the method returns 1.

2)

```
{750,800,850,900,950}

{800,850,900,950,1000}

Returns: 5
```

No matter how the files get moved around, all five drives will still contain data at the end.

3)

```
{49,49,49,49,49,49,49,49,49,49,49,49,49,49,49,49,49,49,49,49,49,49,4
9,49,49,49,49,49,49,49,49,49,49,49,49,49,49,49,49,49,49,49,49,49,49}
```

```
{50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,5
0,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50}
```

```
Returns: 49
```

To consolidate this data, you would select one hard drive, and transfer 1 MB of data from it to each other drive. This results in 49 completely full hard drives and one empty drive.

4)

```
{331,242,384,366,428,114,145,89,381,170,329,190,482,246,2,38,220,290,402,385}
```

```
{992,509,997,946,976,873,771,565,693,714,755,878,897,789,969,727,765,521,961,906
}
```

```
Returns: 6
```