Mock Interviews @ ACM

Problem Card

Room Draw Woes

Problem Statement

You and your room draw group are looking to be housed next year! You're hoping that housing won't mess up this time and forget about any of you like they did last time. Can you help them out?

You have a draw group with `N` people and Housing has a specific set of room sizes (i.e. the number of people each room can house) `rooms` they offer. Write a function `roomDrawOne(N, rooms)` returning the *minimum* number of rooms Housing can assign to your draw group so that everyone is accommodated *and* no room is under capacity. (Assume there is an infinite supply of each room.) If it's impossible to accommodate your draw group exactly, output `-1`.

Assumptions. You can assume `N` and the elements of `rooms` (i.e. possible room sizes) will fit in a 32-bit positive integer.

Example. If `N = 7` and `rooms = {1, 2, 4}`, this would correspond to having a draw group of 7 people and only having singles, doubles, and quads available. In this case, the correct output is 3, because we can accommodate 7 people with one single, one double, and one quad; moreover, this is the assignment that accommodates everyone exactly with the fewest number of rooms. (Note that two quads is an overutilization of resources.)

As usual, you should analyze the runtime and space usage of any solution you come up with.

Test Cases

```
// Input 1
int N = 7; int[] rooms = {1, 2, 4};
System.out.println(roomDrawOne(N, rooms));
```

```
// Output 1
3

// Explanation
// See the example above.

// Input 2
int N = 3; int[] rooms = {2, 4};
System.out.println(roomDrawOne(N, rooms));

// Output 2
-1

// Explanation
// There's no way to house exactly 3 people in only doubles and quads.

// Input 3
int N = 9; int[] rooms = {3, 5};
System.out.println(roomDrawOne(N, rooms));

// Output 3
3
```