We learned the Greedy algorithm last week. So, I chose the problem type is Greedy. The name of this problem is about towns.

The first thing to understand is that this problem is saying about towns with a given population are merged in pairs and named after the original city with a larger population. If the population of the city is the same, either of the two names is chosen. There may be several names left.

Then analyze how to solve the problem. The first is that the current largest city is bound to be the last (eventually named after it). If the remaining urban population and smaller than the current city, it is impossible to eat the largest city, the end of the program; Otherwise the second city might survive.

After analysis, it can help me write code faster. Firstly, I use the Arrays.sort() to sort the array. Like this:

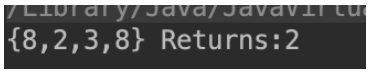
```
Arrays.sort(population);
```

Next, it is worth noting that sum defines the data type as long.
Then, I use two For loop and if statement. Like this:

```
for(int i = 0 ; i<population.length; i++){
    sum = sum+population[i];
}
for(int i= population.length -1; i>= 0; i--) {
    result++;
    sum = sum - population[i];
    if (sum<population[i])
        break;
}
```

Class: SlimeXSlimesCity
Test passed.
Because I can't upload it to web submission, the example is shown below and I have tested the results to be consistent with the example.

```
2)
   {8,2,3,8}
   Returns: 2
   There may be multiple towns with the same population.
3)
   {1000000000, 999999999, 999999998, 999999997}
   Returns: 3
4)
   {1,1,1}
   Returns: 3
5)
   {1, 2, 4, 6, 14, 16, 20}
   Returns: 3
```



The challenge of this problem is that I have to be careful about overflow.

https://community.topcoder.com/stat?c=problem_statement&pm=11154