

假设你是一位很棒的家长，想要给你的孩子们一些小饼干。但是，每个孩子最多只能给一块饼干。对每个孩子  $i$ ，都有一个胃口值  $g_i$ ，这是能让孩子满足胃口的饼干的最小尺寸；并且每块饼干  $j$ ，都有一个尺寸  $s_j$ 。如果  $s_j \geq g_i$ ，我们可以将这个饼干  $j$  分配给孩子  $i$ ，这个孩子会得到满足。你的目标是尽可能满足越多数量的孩子，并输出这个最大数值。

**注意：**

你可以假设胃口值为正。

一个小朋友最多只能拥有一块饼干。

**示例 1：**

输入：[1,2,3], [1,1]

输出：1

**解释：**

你有三个孩子和两块小饼干，3个孩子的胃口值分别是：1,2,3。

虽然你有两块小饼干，由于他们的尺寸都是1，你只能让胃口值是1的孩子满足。

所以你应该输出1。

**示例 2：**

输入：[1,2], [1,2,3]

输出：2

**解释：**

你有两个孩子和三块小饼干，2个孩子的胃口值分别是1,2。

你拥有的饼干数量和尺寸都足以让所有孩子满足。

所以你应该输出2。

先把小孩的胃口和饼干的大小排序，然后进行遍历，如果饼干大小，满足一个小孩胃口，胃口就 +1，饼干依次递增

```
1 def findContentChildren(self, g, s):
2     """
3     :type g: List[int]
4     :type s: List[int]
5     :rtype: int
6     """
7     g = sorted(g)
8     s = sorted(s)
9     gi = si = 0
10    while gi < len(g) and si < len(s):
11        if g[gi] <= s[si]:
12            gi += 1
```

```
13         si += 1
14     return gi
```