1. 题目
2. 题干

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]，这是能让孩子们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] >= 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。

1. 示例

示例 1:

输入: g = [1,2,3], s = [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: g = [1,2], s = [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.

提示：

1 <= g.length <= 3 \* 104

0 <= s.length <= 3 \* 104

1 <= g[i], s[j] <= 231 - 1

1. 题解
2. 思路

从题目中可以看出，我们需要将饼干分给合适的人，什么叫做合适的人呢？就是在能够发给这个人的所有饼干的尺寸中选取一个最小的尺寸发给他，这样，就能将更大的尺寸的饼干留给更需要的人，也就能够保证有更多的人获得饼干。那么，我们需要将饼干的尺寸数组进行排序，同样地也需要将孩子们的胃口数组进行排序。按照从小到大的顺序， 为从小到大的胃口匹配最合适的饼干尺寸，为两个数组分别设置一个指针，如果该饼干能够匹配该胃口，则两个指针都后移；如果该饼干不能匹配该胃口，说明需要更大尺寸的饼干，则饼干指针后移，胃口指针不动；如此迭代，直到到达饼干数组的末尾，这说明所有的饼干都比较完了，这样胃口指针所指的位置就是最后一个能够满足的孩子。

1. 代码实现

Java：

class Solution {

    public int findContentChildren(int[] g, int[] s) {

        Arrays.sort(g);

        Arrays.sort(s);

        int i = 0, j = 0;

        int sum = 0;

        while(i < s.length && j < g.length){

            if(s[i] >= g[j]){

                i++;

                j++;

            }else{

                i++;

            }

        }

        return j;

    }

}