

Assign Cookies

Assume you are an awesome parent and want to give your children some cookies. But, you should give each child at most one cookie. Each child i has a greed factor g_i , which is the minimum size of a cookie that the child will be content with; and each cookie j has a size s_j . If $s_j \geq g_i$, we can assign the cookie j to the child i , and the child i will be content. Your goal is to maximize the number of your content children and output the maximum number.

Note:

You may assume the greed factor is always positive.

You cannot assign more than one cookie to one child.

Example 1:

Input: [1,2,3], [1,1]

Output: 1

Explanation: You have 3 children and 2 cookies. The greed factors of 3 children are 1, 2, 3.

And even though you have 2 cookies, since their size is both 1, you could only make the child whose greed factor is 1 content.

You need to output 1.

Example 2:

Input: [1,2], [1,2,3]

Output: 2

Explanation: You have 2 children and 3 cookies. The greed factors of 2 children are 1, 2.

You have 3 cookies and their sizes are big enough to gratify all of the children,

You need to output 2.

Solution 1

```
Arrays.sort(g);
Arrays.sort(s);
int i = 0;
for(int j=0; i<g.length && j<s.length; j++) {
    if(g[i]<=s[j]) i++;
}
return i;
```

Just assign the cookies starting from the child with less greediness to maximize the number of happy children .

written by [fabrizio3](#) original link [here](#)

Solution 2

My solution from the contest:

```
def findContentChildren(self, g, s):  
    g.sort()  
    s.sort()  
    res = 0  
    i = 0  
    for e in s:  
        if i == len(g):  
            break  
        if e >= g[i]:  
            res += 1  
            i += 1  
    return res
```

$O(n \log n)$ time and $O(1)$ space

written by [dalwise](#) original link [here](#)

Solution 3

```
class Solution(object):
    def findContentChildren(self, g, s):
        """
        :type g: List[int]
        :type s: List[int]
        :rtype: int
        """
        g.sort()
        s.sort()

        childi = 0
        cookiei = 0

        while cookiei < len(s) and childi < len(g):
            if s[cookiei] >= g[childi]:
                childi += 1
            cookiei += 1

        return childi
```

written by [vikram4](#) original link [here](#)

From [LeetCoder](#).