455 Assign Cookies

2018年4月11日 19:12

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 $s_j >= g_i$, 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.

来自 <https://leetcode.com/problems/assign-cookies/description/>

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。

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。

来自 <https://leetcode-cn.com/problems/assign-cookies/description/>

Solution for Python3:

```
class Solution:
    def findContentChildren(self, g, s):
```

```python
        """
        :type g: List[int]
        :type s: List[int]
        :rtype: int
        """
        g.sort()
        s.sort()
        i, j= 0, 0
        while i < len(g) and j < len(s):
            if s[j] >= g[i]:
                i += 1
            j += 1
        return i
```

Solution for C++:

```cpp
class Solution {
public:
    int findContentChildren(vector<int>& g,
vector<int>& s) {
        sort(begin(g), end(g));
        sort(begin(s), end(s));
        int i = 0;
        for (int j = 0; i < g.size() && j
< s.size(); j++) {
            if (g[i] <= s[j]) {
                i++;
            }
        }
        return i;
    }
};
```