Leetcode\_455\_AssignCookies\_Easy

# Leetcode\_455\_AssignCookies\_Easy

## 题目介绍

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 gi, which is the minimum size of a cookie that the child will be content with; and each cookie j has a size sj. If sj >= gi, 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.

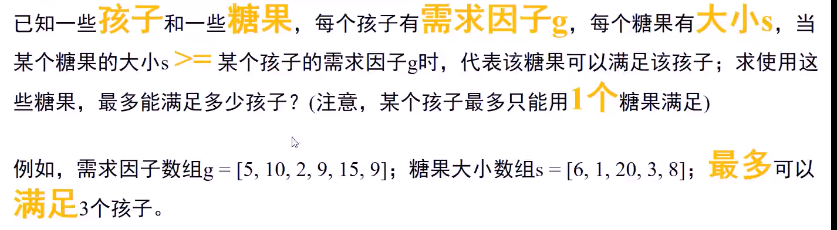
You need to output 1.

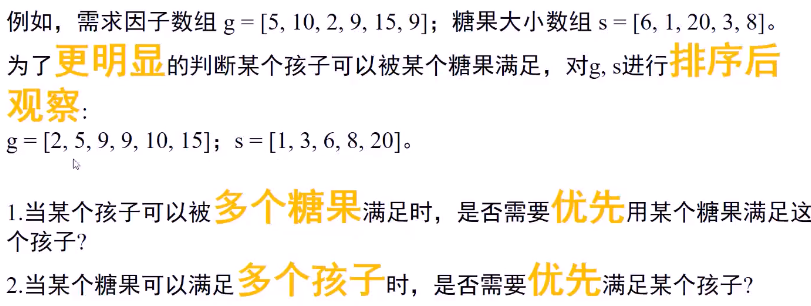
**Example 2:** Input: [1,2], [1,2,3] Output: 2

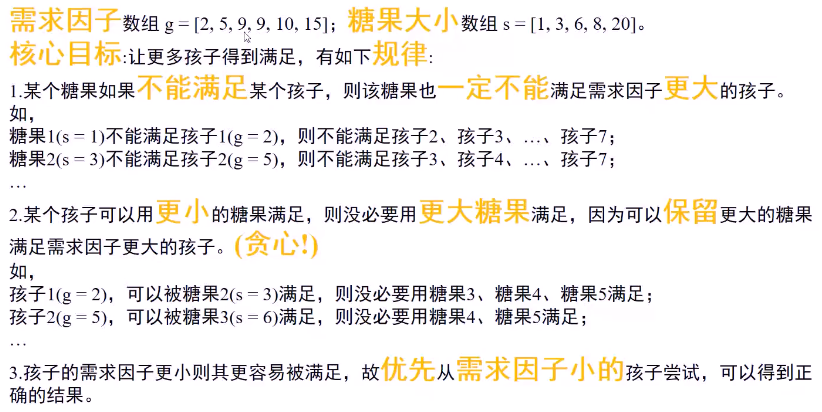
Explanation: You have 2 children and 3 cookies. The greed factors of 2 children are 1, 2.

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

You need to output 2.







## 思路分析

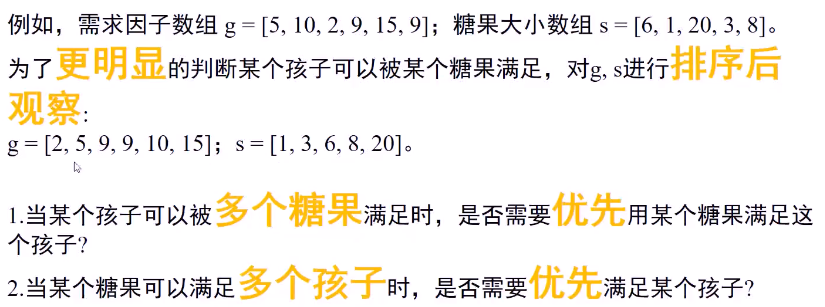
\* 思路分析：首先对g和s排序；

\* 按照**贪心思想**：定义两个指针分别指向g和s的最左端，以小孩需求g为基准从左向右，

\* 按照从左到右的顺序遍历s，若s满足，则结果加1，gIndex和sIndex都加1，

\* 否则直接抛弃s中该糖块，sIndex加1。

\* 因为小糖块不满足当前需求低的小孩，更不可能满足需求高的小孩了。



## Java代码

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

if(g == null||s == null||g.length == 0 || s.length == 0) return 0;

int result = 0;

//对需求g和糖块s排序

Arrays.sort(g);

Arrays.sort(s);

//两个指针

int gIndex = 0,sIndex = 0;

while(gIndex < g.length && sIndex < s.length){

if(s[sIndex] >= g[gIndex]){//该糖块满足需求

result++;

sIndex++;

gIndex++;

}else{//当前糖块不满足该小朋友需求

sIndex++;

}

}

return result;

}

