Chat with the Page

CS23331-DAA-2024-CSE / 2-G-Cookies Problem

2-G-Cookies Problem

Started on	Friday, 29 August 2025, 1:49 PM
State	Finished
Completed on	Sunday, 31 August 2025, 3:27 PM
Time taken	2 days 1 hour
Marks	1.00/1.00
Grade	10.00 out of 10.00 (100 %)

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.

Input:

3 123

2

Output:

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.

Constraints:

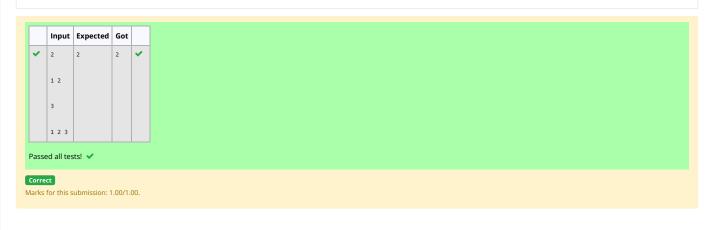
1 <= g.length <= 3 * 10^4

0 <= s.length <= 3 * 10^4

1 <= g[i], s[j] <= 2^31 - 1

Answer: (penalty regime: 0 %)

```
int compare(const void *a, const void *b) {
   return (*(int*)a - *(int*)b);
   ...
  7
8 int findContentChildren(int* g, int gSize, int* s, int sSize) {
9    qsort(g, gSize, sizeof(int), compare);
10    qsort(s, sSize, sizeof(int), compare);
11    int contentChildren = 0;
                  for(int i=0;i<gSize;i++){</pre>
                          for(int j=0;j<sSize;j++){
    if(s[j]>=g[i]){
                                           contentChildren++;
break;
20
21
         int main() {
   int gSize, sSize;
                 scanf("%d", &gSize);
int *g = (int*)malloc(gSize * sizeof(int));
for (int i = 0; i < gSize; i++) {
    scanf("%d", &g[i]);</pre>
                 scanf("%d", &sSize);
int *s = (int*)malloc(sSize * sizeof(int));
for (int i = 0; i < sSize; i++) {
    scanf("%d", &s[i]);</pre>
                 int result = findContentChildren(g, gSize, s, sSize);
printf("%d\n", result);
```



Finish review

Back to Course