pkrc267 [1] Log Ou

questions tags users badges unanswered ask a question about faq

CodeChef Discussion

Search Here...

● questions ○ tags ○ users

RESCALC - Editorial

0

PROBLEM LINK:

Practice
Contest

Author: Andrii
Tester: Praveen Dhinwa
Editorialist: Ajay K. Verma

DIFFICULTY:

Easy

PREREQUISITES:

Ad-hoc, Basic maths.

PROBLEM:

For a given list, the score is defined as the sum of size of the list and an additional number which could be 0, 1, 2, or 4 depending of whether the list has fewer than 4 unique elements, exactly 4 unique elements, exactly 5 elements or exactly 6 unique elements. Given a collection of lists, find the list with highest score, or indicate if there is a tie.

QUICK EXPLANATION:

Compute the score of each list, and return the one with the highest score, or indicate that there is a tie.

EXPLANATION:

In order to solve this problem, one only needs to compute the score of a list efficiently. If we can do so, then the rest of the problem is easy. Compute the score of all lists, and if there is a single list with the highest score, returns its id. On the other hand, if there are multiple lists with the highest score, then indicate that there is a tie.

The score of a list is the sum of its size and an additional number, which depends on the number of unique elements in the list. Since size of a list is easily computable, we only need to compute the number of unique elements in the list.

One could use a set to compute the number of unique elements in the list. However, since all the number are in the range [1, 6], we can use an integer to represent the set of elements in the list.

```
int CountUnique(vector<int> v) {
  int signature = 0;
  for (int x : v) {
    signature |= (1 << x);
  }
  return __builtin_popcount(signature);
}
```

For the list we use an integer signature such that the i-th bit of signature is set iff the integer i is in the input list. In order to count the number of unique elements in the list, one only needs to count the number of bits in signature which are set to 1. This can be done using the __builtin_pop_count() function. One could also precompute these values, and use them because the in this problem the signature will always lie in the range [1, 127].

TIME COMPLEXITY:

O (N), where N is the sum of lengths of all lists.

AUTHOR'S AND TESTER'S SOLUTIONS:

Author's solution
Tester's solution

basic-math editorial ad-hoc rescalc bitset easy sept16

This question is marked "community wiki".

edited **23 hours ago**
admin ♦♦
[12.5k]●346●481●497

asked **yesterday**
djdolls
[2.0k]●50●80●80
accept rate: 0%

Follow this question
By Email:
You are not subscribed to this question.

subscribe me

(you can adjust your notification settings on your profile)

By RSS:
Answers

Answers and Comments

Tags:

editorial **×7,521**

easy **×1,718**

ad-hoc **×466**

sept16 **×85**

basic-math **×84**

bitset **×28**

rescalc **×4**

Asked: **yesterday**

Seen: **1,458 times**

Last updated: **16 hours ago**

Related questions

LEXOPAL - Editorial

PPNUM - Editorial

CHEFFED - Editorial

BALLS - Editorial

SETDIFF - Editorial

PRLADDU-Editorial

Editorial - TRIANGCL

FSTSQ - Editorial

SWEGARR - Editorial

CHAPD - Editorial

6 Answers:

oldest newest most voted

2

This editorial is not good. It should explain it in an easy way.

link | award points

answered **yesterday**

easy_
[101]●2
accept rate: 0%

2

Simplest approach:

1.) Store the frequencies of the cookies.

2.) Sort it

3.) Take the smallest, subtract from all and multiply by 4. This will make all pairs of 6.

4.) Do the 3rd step for 2 more times changing the value as specified. (for pairs of 5 and 4)

5.) Just add the number of cookies to your final answer and you are done!

PS: Just check for a tie and Chef's winning!

AC Solution with above approach:
https://www.codechef.com/viewsolution/11305714

link | award points

answered **22 hours ago**

rishabh.jain9196
[209]●6
accept rate: 0%

I did the same. https://www.codechef.com/viewsolution/11395816

easy_ (16 hours ago)

0

solution
approach:
type is an array storing the count of 6 type of cookies. sort it.
points gained by that person will be 2*type[0]+type[1]+type[2]+number of cookies,c[i].

link | award points

answered **yesterday**

w_white
[1]
accept rate: 0%

0

complex editorial! I understand that this problem is above my level, but then this editorial is way way above. It doesnt explain the main part in depth. can someone point to a better and understandable version?

link | award points

answered **yesterday**

saikiran_skr
[1]
accept rate: 0%

0

Can anyone Help me! My solution is working for only 20pnts. https://www.codechef.com/viewsolution/11459490

link | award points

answered **yesterday**

nanhe
[11]●2
accept rate: 0%

0

This editorial doesn't explain the main part. It explains till the part where we have the number of unique elements in the list. But there are steps afterwards also like calculating the number of boxes possible of different types. That part isn't explained in this editorial.

link | award points

answered **21 hours ago**

kishu18_agar
[21]
accept rate: 0%

Your answer

[hide preview] □ community wiki

Type the text

Privacy & Terms

Post Your Answer

About CodeChef | About Directi | CEO's Corner
CodeChef Campus Chapters | CodeChef For Schools | Contact Us

Powered by OSQA