

← Practice Programming Problems / K-Distribution

K-Distribution

● Submissions Attempted by: 211 | Solved by: 73 | Partially Solved by: 122 | ★★★☆

Problem Editorial My Submissions Analytics

¶ June Clash '15

In order to get at least some points you can try **all possible distributions** and for every of them check if requirements are satisfied. From *Constraints* section it is obvious that given solution should score at least **28 points**. But to get full score you need different, faster approach.

In problem *K-distribution* you are asked about number of **valid partitions** however, as it often happens in such tasks, it is easier to approach it from an opposite such Let's count **bad partitions** – answer to original problem will be equal to total number possible partitions minus number of bad partitions.

Counting bad partitions turns out to be a **standard DP problem** – it's one of modifications of **knapsack problem**. State of DP: *dp[sum]* – *number of ways to pick numbers which sum up to S*. Add numbers one by one and recalculate values of DP (read about knapsack problem first if you are not familiar with it). When calculating final answer, don't forget that two symmetric ways of distribution are counted as different.

Also you have to handle cases when both groups will have sum **less than K**. Depending on implementation, you may count them twice or not count at all. Look at solutions below to see how it can be handled (again, with some sort of **inclusion-exclusion idea**). However, it turns out that things are much simpler. When both groups have sum **less than K** – it means that total sum of all numbers is **less than 2*K**. But in this case answer will be simply **0**, and you don't even need to calculate values of DP table.

Edit Editorial

IS THIS EDITORIAL HELPFUL?



Yes, it's helpful



No, it's not helpful

14 developer(s) found this editorial helpful.

Author Solution by Pavel Sheftelevich

```
1. #include <cstdio>
2. #include <algorithm>
3. #include <vector>
4. #include <iostream>
5. #include <set>
6. #include <map>
7. #include <iomanip>
8. #include <string>
9. #include <string.h>
10. #include <cstdlib>
11. #include <bitset>
12. #include <cmath>
13.
14. #define X first
15. #define Y second
16. #define mp make pair
17. #define pb push back
18.
19. typedef long long ll;
20.
21. using namespace std;
23. const int MAXN = 110, MOD = 1E9 + 7, MAXK = 100100;
24. int n, a[MAXN], k;
25. int dp[MAXK];
26. ll totSum ;
27. int main() {
28.
        cin>>n>>k;
29.
        for (int i = 0; i < n; i++) {
30.
             cin>>a[i];
31.
             totSum += a[i];
32.
        }
33.
        dp[0] = 1;
34.
        for (int i = 0; i < n; i++) {
35.
             if (a[i] >= k) {
36.
                 continue;
37.
38.
             for (int j = k - a[i]; j >= 0; j--) {
39.
                 dp[j + a[i]] += dp[j];
40.
                 if (dp[j + a[i] ] >= MOD) {
41.
                      dp[j + a[i]] -= MOD;
42.
                 }
             }
43.
44.
        }
        int sum = 0;
45.
46.
        for (int i = 0; i < k; i++) {
47.
             //cerr<<i<" : "<<dp[i]<<endl;
48.
             sum += dp[i];
49.
             if (sum >= MOD) {
50.
                 sum -= MOD;
51.
             }
52.
53.
             if (i != totSum - i) {
```

```
//cerr<<"fklfl"<<endl;</pre>
54.
55.
                   sum += dp[i];
56.
                   if (sum >= MOD) {
57.
                        sum -= MOD;
58.
                   }
59.
              }
        }
60.
61.
        //cerr<<sum<<endl;
        int tot = 1;
62.
63.
        for (int i = 0; i < n; i++) {
             tot *= 2;
64.
65.
              if (tot >= MOD) {
                   tot -= MOD;
66.
67.
              }
68.
        }
        tot -= sum;
69.
70.
        if (tot < 0) {
71.
             tot += MOD;
72.
         }
73.
        cout<<tot<<endl;</pre>
74.
        return 0;
75. }
```

Tester Solution by Bohdan Pryshchenko

```
1. #include<bits/stdc++.h>
 2. #define bs 1000000007
 using namespace std;
 6. int n,a[1<<20],ans,T;
 7. int answ;
 8. int dp[1<<20];
 9. long long S;
10. long long rem;
11.
12. void add(int &a,int b)
13. {
14.
            a+=b;
            if (a>=bs)
15.
16.
                     a-=bs;
17. }
18.
19. int main(){
20.
21. cin>>n;
22. cin>>T;
23. --T;
24.
25. for (int i=1;i<=n;i++)
26. cin>>a[i];
27.
28. ans=1;
```

```
29. for (int i=1;i<=n;i++)
30.
   ans*=2.
31.
    ans%=bs;
32.
33. S=0;
34. for (int i=1;i<=n;i++)
35.
             S+=a[i];
36.
37. dp[0]=1;
38. for (int i=1;i<=n;i++)
39.
             for (int j=T-a[i]; j>=0; -- j)
40.
                      if (dp[j]>0)
41.
                               add(dp[j+a[i]],dp[j]);
42.
43. answ=0;
44. for (int i=0; i <=T; i++)
45. {
46.
             rem=S-i;
             if (rem<=T)</pre>
47.
48.
                      add(answ,dp[i]);
49.
             else
50.
                      add(answ,2*dp[i]%bs);
51. }
52.
53. ans=(ans-answ+bs)%bs;
54. cout<<ans<<endl:
55.
56. return 0;}
```

COMMENTS (4)





Join Discussion...

Cancel Post



nagasiva sudhakarreddykovvuri 4 months ago

whyad add (answ,2*dp[i]%s);

Reply ● Message ● Permalink



Bohdan Pryshchenko 4 months ago

You need to count every distribution twice, because if you have some way to pick items with cost X from set with total cost S, it gives you 2 distributions: (X,S-X) and (S-X,X) - you may choose to put your items in either first or second part.

Reply • Message • Permalink



nagasiva sudhakarreddykovvuri 4 months ago

what abt add(answ,dp[i]);

Reply • Message • Permalink



Bohdan Pryshchenko 4 months ago

These are cases which will be counted only once, so you don't need *2 for them.

Reply • Message • Permalink

PROFILE IMPACT

Complete Profile

*Excellent profile will increase your profile discoverability and keep you on top among others.

PROBLEMS SUGGESTED FOR YOU

The function problem

Solved by 249

Game Of Strengths

Solved by 497

Putting matrix

Solved by 17

more...

RECENT SUBMISSIONS

User	Result	Time	Lang
Swapnil		2.7472	C++
Roman Bi		1.7096	С
Himadri		13.4437	C++
Himadri		13.4778	C++
Parag Go		1.7106	C++
Parag Go		1.71	C++
Parag Go		2.2292	C++
View All			

TRENDING NOTES

Number Theory - III

written by Boris Sokolov

Exact String Matching Algorithms

written by Alei Reyes

Binary Indexed Tree or Fenwick Tree

written by Chandan Mittal

Small tricks in for loop

written by Rangeesh

Strings And String Functions

written by Vinay Singh

more ...

DEVELOPERS TO FOLLOW



সায়ান চৌধুরী 2160 followers



Hemesh Singh 6 followers



Sachin Gupta 6361 followers

COMPANIES TO FOLLOW

Gigstart

1717 followers

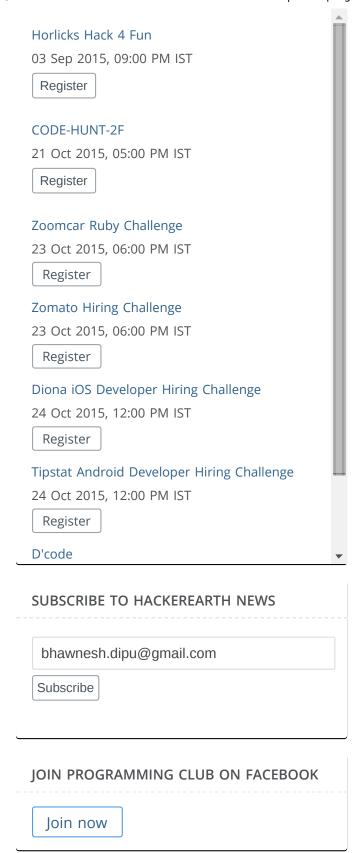
Zoomcar

552 followers

HARMAN

1971 followers

RECOMMENDED CHALLENGES



ABOUT US	HACKEREARTH	DEVELOPERS	
Blog	API	AMA	
Engineering Blog	Chrome Extension	Code Monk	

K-Distribution | Solve programming problems on HackerEarth

Updates & Releases

Team

Careers

In the Press

CodeTable

HackerEarth Academy

Developer Profile

Resume

Campus Ambassadors

Get Me Hired

Privacy

Terms of Service

Judge Environment

Solution Guide

Problem Setter Guide

Practice Problems

HackerEarth Challenges

College Challenges

RECRUIT

Developer Sourcing

Lateral Hiring

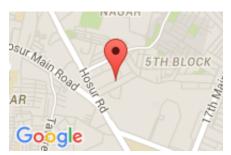
Campus Hiring

FAQs

Customers

Annual Report

REACH US



Illrd Floor, Salarpuria Business Center, 4th B Cross Road, 5th A Block, Koramangala Industrial Layout, Bangalore, Karnataka 560095, India.

contact@hackerearth.com

+91-80-4155-4695

+1-650-461-4192







 g_{\dagger}



© 2015 HackerEarth