← Practice Programming Problems / Candy Distribution 3

Candy Distribution 3

Submissions Attempted by: 68 | Solved by: 40 | Partially Solved by: 14

Combinatorics Math Medium Edit

LIVE EVENTS 1

Problem Editorial My Submissions Analytics

The problem "Candy Distribution 3" doesn't have any editorial. You can contribute it by sending editorial in markdown format to chandan@hackerearth.com.

Add Editorial

Author Solution by Ashish Khatkar

```
/*
ID: ashish1610
PROG:
LANG: C++
*/
#include<bits/stdc++.h>
using namespace std;
#define ll                          long long int
#define vi                          vector<int>
#define vl                          vector<ll>
#define pii                         pair<int,int>
#define pil                         pair<int, ll>
#define pll                         pair<ll, ll>
#define pli                 pair<ll, int>
#define pb(v, a)            v.push_back(a)
#define mp(a, b)            make_pair(a, b)
#define MOD                         1000000007
#define rep(i, a, b)    for(i=a; i<=b; ++i)
#define rrep(i, a, b)   for(i=a; i>=b; --i)
#define si(a)               scanf("%d", &a)
#define sl(a)               scanf("%lld", &a)
#define pi(a)               printf("%d", a)
#define pl(a)               printf("%lld", a)
#define pn                          printf("\n")
ll pow_mod(ll a, ll b)
{
        ll res = 1;
        while(b)
        {
                if(b & 1)
                        res = (res * a) % MOD;
```

```
                a = (a * a) % MOD;
                b >>= 1;
        }
        return res;
}
ll pow2[1000005], pow3[1000005];
void pre_compute()
{
        int i;
        pow2[0] = 1;
        pow3[0] = 1;
        rep(i, 1, 1000000)
        {
                pow2[i] = (2 * pow2[i - 1]) % MOD;
                pow3[i] = (3 * pow3[i - 1]) % MOD;
        }
}
int main()
{
        pre_compute();
        int t, i, n;
        si(t);
        ll ans;
        rep(i, 1, t)
        {
                si(n);
                ans = (pow2[n] * pow2[n]) % MOD;
                ans -= 2 * pow3[n];
                while(ans < 0)
                        ans += MOD;
                if(ans >= MOD)
                        ans = ans % MOD;
                ans += pow2[n];
                if(ans >= MOD)
                        ans = ans % MOD;
                pl(ans);
                pn;
        }
        return 0;
}
```

Tester Solution by Akash Agrawall

```
/******************
        Akash Agrawall
        IIIT HYDERABAD
        **********************/


#include<cstdio>
#include<iostream>
#include<cstdlib>
#include<cmath>
```

```
#include<stdio.h>
#include<string.h>
#include<cassert>
using namespace std;
#define FOR(i,a,b) for(i= a ; i < b ; ++i)
#define rep(i,n) FOR(i,0,n)
#define pln(n) printf("%lld\n",n)
#define sl(n) scanf("%lld",&n)
#define mod (int)(1e9 + 7)
#define ll long long int
ll modpow(ll a,ll n,ll temp){ll res=1,y=a;while(n>0){if(n&1)res=(re
inline ll checkit(ll n)
{
        while(n<0)
                n+=mod;
        if(n>=mod)
                n%=mod;
        return n;
}
ll mod2[1000006],mod3[1000006];
//The final formula after solving the combinatorics is (2^n-1)(2^n
int main()
{
        ll c1,c2,t,n,i;
        mod2[0]=1;
        mod3[0]=1;
        FOR(i,1,1000004)
        {
                mod2[i]=2*mod2[i-1];
                mod3[i]=3*mod3[i-1];
                mod2[i]=checkit(mod2[i]);
                mod3[i]=checkit(mod3[i]);
        }
        sl(t);
        while(t--)
        {
                sl(n);
                c1=mod2[n]-2;
                c1=checkit(c1);
                c2=mod2[n]+1;
                c2=checkit(c2);
                c1=c1*c2;
                c1=checkit(c1);

                c2=mod3[n]-mod2[n]-1;
                c2=checkit(c2);
                c2*=2;
                c2=checkit(c2);

                c1-=c2;
                c1=checkit(c1);
                pln(c1);
        }
        return 0;
```

```
65. }
```

COMMENTS (4)

Refresh

Join Discussion...

Cancel Post

Tarun Dutt 9 months ago

Can you please explain how the formula was obtained?

Reply • Message • Permalink

ankur malik 8 months ago

yes.. Please someone add some link or something, where this thing is explained

Reply • Message • Permalink

artisticoder 2 months ago

A possible answer can be:
A and B cannot contain neither 0 nor n elements (otherwise conditions does not hold). If A contains k elements, then B can be any subset of n elements apart the subset with n and 0 elements, i.e. 2^n - 2, minus the subsets of A apart the empty set, i.e. 2^k - 1, minus all subsets containing the k elements, that is all subsets with n-k elements, apart from subsets with n-k and 0 elements, i.e. 2^(n-k) - 2. So with A having k elements, the total ways to have A and B are C(n, k)*(2^-2 - (2^k-1) - (2^n-k - 2)) = C(n,k)*(2^n - 2^k - 2^(n-k) +1). If the range of k is 1 <= k <= n-1 then we have to sum the above formula over this range. Let's break the formula into three parts. First is Sum(C(n,k)*(2^n + 1)) = (2^n + 1)*(2^n - 2). (Sum(Cn,k) = 2^n by definition, the minus 2 is because the range of k does not include n and 0, for which C is 1 for each one). The second part is Sum(C(n,k)*2^k) which by definition of the binomial coefficient is (1 + 2)^n - 1 - 2^n. The minus is again for excluding 0 and n from k. The third part Sum(C(n, k)*2^(n-k)) is equal to the second one because C(n, k) = C(n, n-k).
Thus, the final formula is:
(2^n + 1)*(2^n - 2) - 2*(3^n - 2^n -1). In the comments of the 2nd solution the formula has a small error, but the code computes the result correctly.

Reply • Message • Permalink

himanshu pal 2 months ago

Let A be a set containing k elements (k is non zero quantity and cannot be equal to n as per the condition given in question)
Now nCk is the number of ways of choosing this set A. Now number of ways of choosing B = T - b⊆a - a⊂b where
T(Total number of sets ignoring null and complete set ie 2^n -2)
b⊆a(case when B is a subset of A ie 2^k-1 ignoring null set "note: we are not ignoring complete set of k element because b can have all elements of k")
a⊂b(case when A is a subset of B ie all k elements already present so 2^(n-k)-1(because if B does not contain any extra element other than k elements that ie B=A we have considered that case in 2nd part b⊆a) -1(because if B contains all elements of n=k+n-k than B becomes complete set which we have to ignore))
So the complete formula of B becomes (2^n - 2 - (2^k - 1) - (2^(n-k) - 2)).Now 1<=k<=n-1 we have to sum the series nCk*(2^n - 2 - (2^k - 1) - (2^(n-k) - 2)). Now by using binomial series Sum(nCk) = 2^n-2 , Sum(nCk*(2^k)) = 3^n - 1 - 2^n = Sum(nCk*(2^(n-k))

because C(n, k) = C(n, n-k) so the whole formula reduces to (2^n-2)(2^n+1)-(2*(3^n - 1-2^n)) which is simplified further to 2^(2n) +2^n -2*(3^n)

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

Most Frequent
Solved by 299

Flip the words
Solved by 17

Majority
Solved by 123

more...

RECENT SUBMISSIONS

User	Result	Time	Lang
MUKESH k...		1.0408	C++
Amir Nas...		1.0213	C++
Amir Nas...		0.9807	C++
Kadumuri...		6.008	C
Kadumuri...		6.0066	C
Sumit Ku...		1.095	C++
Sumit Ku...		1.1618	C++

View All

TRENDING NOTES

Technique to play online Bot games
written by Catalin Stefan Tiseanu

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

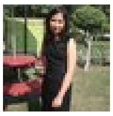
Small tricks in for loop
written by Rangeesh

more ...

DEVELOPERS TO FOLLOW

Gaurav Chand Katoch
0 followers

srajan dongre
1 followers



Shagun Kush
5 followers

COMPANIES TO FOLLOW

VMware
3076 followers

Multunus Software Private Limited
1800 followers

HARMAN
1975 followers

RECOMMENDED CHALLENGES

Horlicks Hack 4 Fun

03 Sep 2015, 09:00 PM IST

Register

CODE-HUNT-2F

21 Oct 2015, 05:00 PM IST

Register

Zoomcar Ruby Challenge

23 Oct 2015, 06:00 PM IST

Register

Zomato Hiring Challenge

23 Oct 2015, 06:00 PM IST

Register

Diona iOS Developer Hiring Challenge

24 Oct 2015, 12:00 PM IST

Register

Tipstat Android Developer Hiring Challenge

24 Oct 2015, 12:00 PM IST

Register

D'code

SUBSCRIBE TO HACKEREARTH NEWS

bhawnesh.dipu@gmail.com

Subscribe

JOIN PROGRAMMING CLUB ON FACEBOOK

Join now

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

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

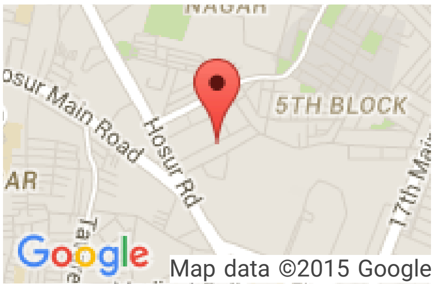
Campus Hiring

FAQs

Customers

Annual Report

REACH US



IIIrd 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