komiyamの日記

<前の5日分

2012-04-01

TCO12 Round1A 500pt : EllysFractions

TopCoder

問題概要

0<a/b<1を満たすような互いに素なa,bで、a*bがn! (1≦n≦N(<100))になっているようなものがいくつあるか求める問題。

解法

積がn!になるようなのを考える。n!=Πp_i^e_i (e_i>0)と書けるとき、既約なのでp_iは全部分子にいくか分母にいくかである。なのでn以下にある素数の数を2の肩に乗せればよい。このとき分子と分母は当然異なるので分子<分母になってるのは分割のちょうど半分になる。

acceptされたコード

```
#include <cstring>
using namespace std;

typedef long long int64;

const int MAX_SIZE = 300;
bool isPrime[MAX_SIZE];

void  seive (){
        memset(isPrime, -1, sizeof(isPrime));
        bool *ps = isPrime;
```

プロフィール

komiyam

競技プログラミングの記録

カレンダー

<< 2012/04 >>

日	月	火	水	木	金	土
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

カテゴリー

TopCoder
Codeforces
AOJ
ICPC
PKU
雑記
ライブラリ
GCJ
UVa
SPOJ
codechef
テンプレート
その他のコンテスト
Virtual Online Contests
コンテストの結果
蟻本
小ネタ
アルゴリズム

```
        ps[0] = ps[1] = false;
        for(int i=2; i*i<MAX_SIZE; i++)if(ps[i
                for(int j=i<<1; j<MAX_SIZE; j+
                        ps[j] = false;
                }
        }
}

#include <stdio.h>

struct EllysFractions {

        int64 getCount(int N) {
                if(N==1){
                        return 0;
                }
               seive ();

                int64 ans = 0;
                int cnt = -1;
                for(int i=2; i<=N; i++){
                        if(isPrime[i]){
                                cnt++;
                        }
                        ans += 1LL<<cnt;
                }

                return ans;
        }

};
```

Permalink | コメント(0) | トラックバック(0) | 09:25

2012-01-31

Hacker Cup 2012 round 1 : Checkpoint

その他のコンテスト

問題概要

整数S(<10^7)が与えられる。

ライブラリ検証問題

AtCoder

最新タイトル

[TopCoder]TCO12 1B 250pt : FoxAndKgram

[PKU][蟻本]POJ-1740 : A New Stone Game

[AOJ]AOJ-2268, UTPC2011-J : Randomized Self-Balancing Binary Search Tree

[AtCoder]AtCoder Regular Contest #001 D : レースゲーム

[AtCoder]AtCoder Regular Contest #001 C : パズルのお手伝い

[AtCoder]AtCoder Regular Contest #001 B : リモコン

[AtCoder]AtCoder Regular Contest #001 A : センター採点

[TopCoder]SRM 487 250pt : BunnyComputer

[PKU][UVa][蟻本]POJ-1486, LiveArchive-5634, UVa-663, ZOJ-1197, TJU-1611 : Sorting Slides

[TopCoder]SRM 540 550pt : RandomColoring

最近のコメント

2011-02-08 KenjiH

2011-03-04 komiyam

2011-03-04 slipstak2

2011-03-04 slipstak2

2011-09-14 Egtra

最近のトラックバック

2011-06-27 minus9dの記録 - [codeforces]Round #103 (Div. 2)

2011-12-10 k_operafanのTopCoder日記 - 提出してしまうとWrong Answerになって...

2011-12-12 aizuzia - DPの話 (追記)

comb(x+y,y)*comb(a+b,b) = Sとなるような a,b,x,y(a+b>0, x+y>0)で、x+y+a+bのとりうる最小値を求める問題。

解法

comb(x+y,y)がS以下となる組み合わせは多くないので全列挙する。以下のコードは色々怪しい。

acceptされたコード

```
#include <cstdio>
#include <cstring>
#include <algorithm>
using namespace std;

typedef long long int64;

const int MAX_S = 1e7;
const int INF = 1<<29;

int S;
int brute[MAX_S+1];
int mini[MAX_S+1];
int ans[MAX_S+1];
int64 comb[4000][4000];

const int MAX_SIZE = MAX_S+10;
bool isPrime[MAX_SIZE];

void  seive (){
        memset(isPrime, -1, sizeof(isPrime));
        bool *ps = isPrime;
        ps[0] = ps[1] = false;
        for(int i=2; i*i<MAX_SIZE; i++)if(ps[i]
                for(int j=i<<1; j<MAX_SIZE; j+=
                        ps[j] = false;
                }
        }
}


void init(){
        scanf("%d", &S);
```

2011-07-28 aizuzia - DPの話 (追記)

2011-05-20 kanetaiの二次記憶装置 - リンク

最近書いたコメント

2012-03-21 id:simezi_tan:20120318

2011-11-29 id:komiyam:20110304:129916

2011-06-28 id:komiyam:20110619:130844

2011-05-24 id:komiyam:20110206:129696

2011-05-24 id:komiyam:20100711:127883

ページビュー

204820

```
}

int solve(){
        int ans = S+1;
        for(int i=1; i*i<=S; i++)if(S%i == 0){
                int x = (isPrime[i] ? i : mini[
                int y = (isPrime[S/i] ? S/i : m
                ans = min(ans, x+y);
        }
        return ans;
}

void pre(){
       seive ();
        fill(mini, mini+MAX_S+1, INF);
        for(int i=2; i<=MAX_S; i++){
                mini[i] = i;
        }
        mini[1] = 1;
        for(int i=0; i<4000; i++){
                comb[i][0] = comb[i][i] = 1;
                for(int j=1; j<i; j++){
                        comb[i][j] = comb[i-1][
                        if(comb[i][j] >= INF){
                                comb[i][j] = IN
                        }
                        if(comb[i][j] <= MAX_S)
                                mini[comb[i][j]
                        }
                }
        }

}

int main(){
        pre();
        int T;
        scanf("%d", &T);

        for(int c=1; c<=T; c++){
                init();
                printf("Case #%d: %d¥n", c, sol
        }

        return 0;
}
```

2012-01-11

CodeChef-MISINT2 : Misinterpretation 2

codechef

問題概要

長さnの文字列で、偶数番目+奇数番目と連結した文字列ともとの文字列が一致するのは何通りあるかの和を区間[L,R]について求めよ。L-R<5*10^4, R<10^10。

解法

小さいケースについてはunion-findでつなげて親の個数を数えてやればよい。これで試すと偶数項と奇数項の間に分かりやすい関係が見つかり片方を数列辞典に投げると見事にヒットする。ord_2(n)を2^mが1とmod nで等しくなるような最小の正整数mで定めると、奇数項についてa[n]=Σ_{d|n}phi(d)/ord_2(d)となるらしい。制約が明らかに区間篩を臭わせていて、実際に区間篩を行ってついでに素因数分解しておくことによってphi(d)はdを列挙するのと同時に計算できる。ord_2(Π p_i ^ e_i) = LCM(ord_2(p_i^e_i))が成り立つこと予想できるのでこれを使ってord_2(p_i)もそれなりの速度で計算できる。小さいところでord_2(p^k) = p * ord_2(p^(k-1))がk>=2について成り立つらしいことが確かめられる(証明分からない)のでそれを使って高速化したりする。k=1のときは、phi(p)=p-1を素因数分解(ord_2(n) | phi(n)なので)して愚直に求めて小さいところでメモ化したりしてやると良い。

この問題、ord_2の計算にphi(d)の約数を調べ

るだけでも頑張って高速化すれば想定解の2倍程度の速度までは平気で出るので、必死に高速化頑張ったけど結果的にはそんなに意味はなかった。高速化の練習にはなったと思いたい。あと上限10^10が微妙にいやらしくてオーバーフローに注意する必要もある。

acceptされたコード

```
nclude <cstdio>
nclude <cstring>
nclude <algorithm>
ng namespace std;

edef unsigned long long int64;

st int MAX_RT = 1e5+10;
st int MAX_W = 5e4+10;
st int64 MOD = 1e9 + 7;
st int MEMO_SIZE = 1e5;
st int MAX_SIZE = 1e5+100;
st int MAX_PRIME = 1e4;


64 R, L;
 fs[MAX_W];
64 ps[MAX_W][11];
 ns[MAX_W][11];
64 pps[11];
 es[11];

l visited[MAX_RT][30];
64 memo[MAX_RT][30];

l vis2[MEMO_SIZE];
64 memo2[MEMO_SIZE];

64 bins[40];

l isPrime[MAX_SIZE];
 P;
 primes[MAX_PRIME];
64 step[MAX_PRIME];

 phiF;
```

```
:64 phipps[11];
: phinns[11];

:64 prev_n, mem;

.d  seive (){
      memset(isPrime, true, sizeof(isPrime));
      isPrime[0] = isPrime[1] = false;
      for(int i=2; i*i<MAX_SIZE; i++)if(isPrime[
              for(int j=i<<1; j<MAX_SIZE; j+=i){
                      isPrime[j] = false;
              }
      }
      for(int i=2, st=1e5+10; i<st; i++){
              if(isPrime[i]){
                      step[P] = (int64)i*i;
                      primes[P++] = i;
              }
      }


.d init(){
      scanf("%lld%lld", &L, &R);
      memset(fs, 0, sizeof(fs));


:64 lcm(int64 x, int64 y){
      return x / __gcd(x, y) * y;


:64 pow_binary2(int64 n, int64 mod){
      int64 ret = 1;

      if(mod < (1LL<<32)){
              for(;n;n&=n-1){
                      ret = ret * bins[__builtin_
              }
      }
      else{
              for(;n;n&=n-1){
                      ret = (((ret * (bins[__bui
              }
      }

      return ret;
```

```
:64 pow_binary(int64 x, int64 n, int64 mod = MOD)
    int64 ret = 1;
    for(;n;n>>=1){
            if( n&1 ){
                    ret = ret * x % mod;
            }
            x = x * x % mod;
    }

    return ret;


:64 pow_binary3(int64 x, int64 n, int64 mod){
    int64 ret = 1;

    if(mod < (1LL<<32)){
            for(;n;n>>=1){
                    if( n&1 ){
                            ret = ret * x % mo
                    }
                    x = x * x % mod;
            }
    }
    else{
            for(;n;n>>=1){
                    if( n&1 ){
                            ret = (((ret * (x>>
                    }
                    x = (((x * (x>>16) % mod)<<
            }
    }

    return ret;


:64 ord2pk(int64 p, int k){
    if(p < MAX_RT && visited[p][k]){
            return memo[p][k];
    }

    if(k == 1){
            int64 n = p;

            int64 m = p - 1;
            phipps[0] = 2;
```

```
phinns[0] = __builtin_ctzll(m);
phiF = 1;
m >>= __builtin_ctzll(m);

for(int i=1; step[i]<=m; i++){
        if(m%primes[i] == 0){
                phipps[phiF] = pri
                phinns[phiF] = 0;
                for(;m%primes[i]==
                        phinns[phi
                }
                phiF++;
        }
}
if(m!=1){
        phipps[phiF] = m;
        phinns[phiF] = 1;
        phiF++;
}

bins[0] = 2;
if(n > (1LL<<32)){
        for(int i=1, ed=64-__built
                bins[i] = (((bins[
        }
}
else{
        for(int i=1, ed=64-__built
                bins[i] = bins[i-1
        }
}


int64 phi = p - 1;
int64 ret = phi;

for(int i=0; i<phiF; i++){
        int64 tt = ret;
        for(int j=0; j<phinns[i]; 
                tt /= phipps[i];
        }

        int64 xx = pow_binary2(tt,
        if(xx == 1){
                ret = tt;
                continue;
```

```
                        }
                        for(int j=0; j<phinns[i]-1
                                tt *= phipps[i];
                                xx = pow_binary3(x
                                if(xx == 1){
                                        ret = tt;
                                        break;
                                }
                        }
                }

                if(p < MAX_RT){
                        visited[p][k] = true;
                        return memo[p][k] = ret;
                }
                return ret;
        }

        else{
                int64 ret = ord2pk(p, 1);
                for(int i=0; i<k-1; i++){
                        ret *= p;
                }

                if(p < MAX_RT){
                        visited[p][k] = true;
                        return memo[p][k] = ret;
                }
                return ret;
        }


:64 ord2(int F){

        int64 ret = 1;
        for(int i=0; i<F; i++)if(es[i] > 0){
                ret = lcm(ret, ord2pk(pps[i], es[i
        }

        return ret;


:64 ord(int64 n){
        int l = __builtin_ctzll(n+1);
        int64 m = (n+1)>>l, ret = l;
        for(;m!=1;){
```

```
            l = __builtin_ctzll(n+m);
            m = (n+m) >> l;
            ret += l;
        }

        return ret;


:64 phi_ord(int64 n, int64 phi, int depth){
        if(n >= MEMO_SIZE){
            return phi / ord2(depth);
        }

        if(vis2[n]){
            return memo2[n];
        }
        vis2[n] = true;
        return memo2[n] = phi / ord2(depth);


:64 dfs(int depth, int64 cur, int64 phi, int idx
        if(depth == fs[idx]){
            return phi_ord(cur, phi, depth);
        }

        int64 ans = 0, p = ps[idx][depth], pp=p-1;
        pps[depth] = p;
        for(int i=0, ed = ns[idx][depth]; i<=ed; i
            es[depth] = i;
            if(i==0){
                ans += dfs(depth+1, cur, pp
            }
            else{
                cur *= p;
                ans += dfs(depth+1, cur, pp
                pp *= p;
            }
        }
        return ans;


d check(int64 n){
        int idx = n - L, f = fs[idx];
        int64 m = n;
        for(int i=0; i<f; i++){
            int64 p = ps[idx][i];
```

```
                ns[idx][i] = 0;
                for(;m%p==0;m/=p){
                        ns[idx][i]++;
                }
        }
        if(m!=1){
                ps[idx][f] = m;
                ns[idx][f] = 1;
                fs[idx]++;
        }


:64 calc(int64 n){
        if(prev_n == n){
                return mem;
        }

        if(n&1){
                check(n);
                prev_n = n;
                return mem = dfs(0, 1, 1, n-L);
        }
        return calc(n+1) - 1;


.d prepare(){
        for(int i=0; i<P; i++){
                for(int64 j=max((int64)2, (L+prime
                        ps[j-L][fs[j-L]] = primes[
                        fs[j-L]++;
                }
        }



: solve(){
        int64 ans = 0;

        prepare();
        prev_n = 1LL<<60;
        for(int64 i=L; i<=R; i++){
                ans += pow_binary(26, calc(i));
                if(ans >= MOD){
                        ans -= MOD;
                }
        }
```

```
    return (int)ans;


main(){
    seive();

    int T;
    scanf("%d", &T);
    while(T--){
            init();
            printf("%d¥n", solve());
    }

    return 0;
```

Permalink | コメント(0) | トラックバック(0) | 19:00

2011-12-26

Xmas Contest 2011 A : input

その他のコンテスト

問題概要

正整数をいくつかとってきて、sum A[i] <= 150, prod A[i] = X (mod 10^9+9)となるようなものを出力する問題。ない場合はその旨出力する。

解法

A[i]は合成数より素数をとった方が和を減らせるので得。なのでそのような素数の組み合わせについて全探索する。あまりに愚直な探索だとTLEするので、和が150を越えないことを利用した枝刈りくらいは必要。また、X=1のとき空の数列を返さないよう注意する。

acceptされたコード

```
#include <cstdio>
#include <vector>
#include <numeric>
#include <cstring>
using namespace std;

typedef  long long int64;

const int64 MOD = (int)1e9 + 9;
const int MAX_N = 150;

const int MAX_SIZE = 300;
bool isPrime[MAX_SIZE];
vector<int> primes;
int path[MAX_N];
char col[26*26][10];

void  seive (){
        memset(isPrime, -1, sizeof(isPrime));
        bool *ps = isPrime;
        ps[0] = ps[1] = false;
        for(int i=2; i*i<MAX_SIZE; i++)if(ps[i
                for(int j=i<<1; j<MAX_SIZE; j+
                        ps[j] = false;
                }
        }
}


int X, L;

void init(){
        scanf("%d", &X);
       seive ();
        for(int i=2; i<=MAX_N; i++){
                if(isPrime[i]){
                        primes.push_back(i);
                }
        }

        L = primes.size();

        for(int i=0; i<26; i++){
                for(int j=0; j<26; j++){
                        col[i*26 + j][0] = i +
```

```
                        col[i*26 + j][1] = j +
                }
        }
}

void prn(){
        int ans = 0;
        for(int i=0; i<L; i++){
                ans += primes[i] * path[i];
        }
        printf("%d¥n", ans);

        for(int i=0, p=0; i<L; i++)if(path[i]
                for(int k=0; k<path[i]; k++){
                        for(int j=0; j<primes[
                                puts(col[p]);
                        }
                        p++;
                }
        }
}

bool search(int idx, int64 cur, int sum){

        if(cur == X){
                prn();
                return true;
        }

        if(idx == L){
                return false;
        }

        int64 q = 1;
        if(sum + primes[idx] > MAX_N){
                return false;
        }
        for(int i=0, p=0; sum + p <= MAX_N; p+
                path[idx] = i;
                if(search(idx+1, cur*q%MOD, su
                        return true;
                }
                q = q * primes[idx] % MOD;
        }
        path[idx] = 0;
```

```
        return false;
}

void solve(){
        if(X==1){
                puts("1");
                puts("a");
        }
        else if(!search(0, 1, 0)){
                puts("NO");
        }
}

int main(){
        init();
        solve();

        return 0;
}
```

Permalink | コメント(0) | トラックバック(0) | 18:44

2011-12-08

SRM 526 500pt : PrimeCompositeGame

TopCoder

問題概要

二人でゲームをする。石がN(<5*10^5)個ある。先手は石を1..K個取り除いて残りが素数になるようにしなければならない。後手は石を1..K個取り除いて残りが合成数になるようにしなければならない。石を取り除くことができなくなったら負けになる。両者が最善を尽くしたとき何手で勝負が付くか求める問題。

考えたこと

- タイトルの割に数論っぽくない問題のよ

うなので安心した。
- 最短手数を求めるゲーム木探索は、勝ちのとき(INF - 手数)、負けのとき-(INF - 手数)を評価値にすればnegamaxで行けるんだっけか。
- ナイーブな実装ではO(N*K)でTLEっぽい。とはいえ単に最小値を取り出すだけだからスライド最小値かRMQで計算量落ちる。
- スライド最小値の方が計算量はよいけど端の処理が面倒なのでRMQで書こう。多分間に合うだろう。
    - 今見返すと結構厳しい。
- よし分かったので書こう。ってこれ先手と後手でルール違うんだからnegamaxじゃダメじゃん。
- しかし修正はそんなに難しくはない。DP配列を2本用意すれば良いだけだ。
- RMQも二つあるのでglobalな配列や関数をstructに押し込む。
- 実行したらRE起こす。なぜ?
- しばらく悪戦苦闘した結果stack overflowしていることに気づく。構造体の中にでかい生配列が入ってるから局所変数にするとそりゃおちる。
- 配列をvectorで書き直して何とか実行可能になった。
    - あるいは、global変数にするか。
- 実行すると負けのとき何故か-INF周辺の値が帰ってきて、勝ちのときは値が2だけずれてる。
- 正直時間ないし理由とかどうでもいいのでサンプルが通るように適当に符号や値を修正する。
- 残り2分というところで何とかサンプル通った。最大ケースらしきものだけテストしてから送信。
- まったく自信はなかったけど通っていた。コーナーケースとかにハマらなかったのは運が良かった。

- なんかmultiset使ったり二分探索使ったりしている解答もあるけどよく分かっていない。
    - 追記：multisetの方は自明だった。というかPOJのSliding Windowsでそれやってて TLEもらいつづけていたのを忘れていた。

本番中提出したコード

計算量O(N*log N)。

```
#include <algorithm>
#include <cstring>
#include <vector>
using namespace std;

const int INF = 1<<28;
const int MAX_N = 474747;

int dp[MAX_N+1][2];
bool checked[MAX_N+1][2];

struct PrimeCompositeGame {

        int theOutcome(int N, int K) {

                if(N == 1){
                        return 0;
                }
                seive ();
                RMQ a, b;

                a.init_rmq(N+1);
                b.init_rmq(N+1);

                checked[0][0] = checked[1][0]
                dp[0][0] = dp[1][0] = (INF - 1
                dp[0][1] = dp[1][1] = (INF - 1

                a.update(0, dp[0][0]);
                a.update(1, dp[0][1]);
                b.update(0, dp[1][0]);
                b.update(1, dp[1][1]);
```

```
        for(int i=2; i<N; i++){
                if(isPrime[i]){
                        checked[i][0]
                        dp[i][0] = (IN
                        a.update(i, dp
                }
                else{
                        checked[i][1]
                        dp[i][1] = (IN
                        b.update(i, dp
                }
        }

        for(int i=2; i<=N; i++){
                //先手
                if(!checked[i][0]){
                        int score = b.
                        if(score > 0){
                                int tu
                                dp[i][
                        }
                        else{
                                int tu
                                dp[i][
                        }
                        a.update(i, dp
                }

                //後手
                if(!checked[i][1]){
                        int score = a.
                        if(score > 0){
                                int tu
                                dp[i][
                        }
                        else{
                                int tu
                                dp[i][
                        }
                        b.update(i, dp
                }
        }

        int ans = dp[N][0];
        if(ans > 0){
                return INF - ans - 2;
```

```
                }
                else{
                        return -(INF + ans) +
                }
        }

};
```

Permalink | コメント(0) | トラックバック(0) | 00:01

<前の5日分