Team notebook

Jonathan es muy guapo

November 8, 2019

Contents

1 Búsqueda Binaria

1.1 Infinite String

```
def fpow(a, e):
    if e == 0 or a == 1: return 1
    ans = fpow(a, e//2)
    if (e&1): return ans*ans*a
    else: return ans*ans
def f(x, b):
    aux = 0
    while x>1:
        aux+= (x*fpow(b,x))
        x-=1
    return aux+(x*fpow(b,x))
def geti(b, x):
    l = 0
    r = 31
    while (r-l>0):
        mid = l+(r-l)//2
        if (f(mid,b)>=x): r = mid
        else: l = mid+1
    if l == 0: return 1
    return l
tc = int(input())
while(tc):
    tc-=1
    b,x = [int(i) for i in input().split()]
    seq = geti(b,x)
    fseqm = f(seq-1,b)
    pos = x-fseqm
    post = (x-fseqm)//seq
```

```
    posdt = (pos%seq)
    s = ['a' for i in range(seq)]
    i = len(s)-1
    while i>=0 and post>0:
        s[i] = chr(ord(s[i])+post%b)
        post= post//b
        i-=1
    print(s[posdt])
```

1.2 Numerical Sequence

```
#include <bits/stdc++.h>
#define forn(i, n) for (int i = 0; i<(int)n; ++i)
#define forr(i, t, n) for (int i = t; i<n; ++i)
#define rmod(x, y) (((x%y)+y)%y)
using namespace std;
typedef unsigned int uint;
typedef long long ll;
typedef unsigned long long ull;
typedef pair<int, int> pii;
const int MAXN = 1e9+1;

int q;
ll k;
ll fpow(ll a, int e){
	if (e == 0 or a == 1) return 1;
	ll ans = fpow(a, e/2);
	if (e & 1) return ans*ans*a;
	else return ans*ans;
}
int dig(ll n) {
	int c = 0;
	while (n>0)	{
		c++;
		n/=10;
	}
	return c;
}
ll a(ll n, int d){
	return n+(n-fpow(10, d-1))+1;
}
ll a2(ll n, int d){
	return n-fpow(10, d-1)+1;
}
ll form(ll n, int d){
	ll aux = a(n, d);
	return ((aux-n)*(aux-n+1))/2;
}
ll getdigsn(ll n) {
	int d = dig(n);
	ll curs = n;
	while (d>1)	{
		curs+= a2(n, d);
		d--;
	}
	return curs;
}
ll totdigs(ll n) {
	int digs = dig(n);
	ll curs = (n*(n+1))/2;
	while(digs>1) {
		curs+= form(n, digs);
		digs--;
	}
	return curs;
}

int main()
{
	ios::sync_with_stdio(0);
	cin.tie(0);
	#ifndef ONLINE_JUDGE
		freopen("_input.txt", "r", stdin);
		freopen("_output.txt", "w", stdout);
	#endif
		cin >> q;
		while (q--)	{
			cin >> k;
			ll l = 0, r = MAXN;
			while (r-l>0){
				ll mi = l+(r-l)/2;
				if (totdigs(mi)>=k) r = mi;
				else l = mi+1;
			}
			ll pos = k-totdigs(l-1);
			ll lo = 0, hi = l+1;
			while (hi-lo>0) {
				ll mi = lo+(hi-lo)/2;
```

```
                        if (getdigsn(mi)>=pos) hi = mi;
                        else lo = mi+1;
                }
                ll apos = pos - getdigsn(lo-1) -1;
                string ans = to_string(lo);
                cout << ans[apos] << "\n";
        }
    return 0;
}
```

2 Matemática

2.1 bear and prime numbers

```
#include <bits/stdc++.h>
#define forn(i, n) for (int i = 0; i<(int)n; ++i)
#define forr(i, t, n) for (int i = t; i<n; ++i)
#define rmod(x, y) (((x%y)+y)%y)
using namespace std;
typedef unsigned int uint;
typedef long long ll;
typedef unsigned long long ull;
typedef pair<int, int> pii;
const double PI = acos(-1);
const int MAXN = 1e6;

int n, m;
int a[MAXN+1];
int dp[MAXN+1];
int spf[MAXN+1];
int ds[MAXN+1];

void sieve() {
    spf[0] = spf[1] = 1;
    for (int i=2; i<MAXN; i++)
        if (i&1) spf[i] = i;
        else spf[i] = 2;
    for (ll i=3; i*i<MAXN; i++)
        if (spf[i] == i)
                for (ll j=i*i; j<MAXN; j+=i)
                if (spf[j]==j) spf[j] = i;
}
```

```
void logpd(int x) {
        int s;
        while (x>1)    {
                s = spf[x];
                ds[s]++;
                while (s>1 and x>1 and not (x%s)) x/=s;
        }
}
int main(){
        ios::sync_with_stdio(0);
        cin.tie(0);
        sieve();
        cin >> n;
        forn(i, n) {
                cin >> a[i];
                logpd(a[i]);
        }
        dp[0] = dp[1] = 0;
        forr(i, 2, MAXN) dp[i] = dp[i-1] + ds[i];
        cin >> m;
        int l,r ;
        while (m--)    {
                cin >> l >> r;
                l = min(l, 10000001);
                r = min(r, 10000001);
                cout << (dp[r] - dp[l-1]) << "\n";
        }
        return 0;
}
```

2.2 Fibonacci con matrices

```
#include <bits/stdc++.h>
#define forn(i, n) for (int i = 0; i<(int)n; ++i)
#define forr(i, t, n) for (int i = t; i<=n; ++i)
#define rforr(i, t, n) (int i = n-1; i>=t; --i)
#define rfor(i, n) for (int i = n-1; i>=0; --i)
#define rmod(x, y) (((x%y)+y)%y)
#define pb push_back
#define emp emplace_back
using namespace std;
typedef unsigned int uint;
typedef unsigned short ushort;
```

```
typedef long long ll;
const int MAXN = 1024;

//OPERATORS


//GLOBALS


//FUNCTIONS
void multiplicar(int (&a)[2][2], int b[2][2]){
	int res[2][2];
	for(int i = 0; i < 2; i++){
		for(int j = 0; j < 2; j++){
			res[i][j] = 0;
			for(int k = 0; k < 2; k++){
				res[i][j] += ((a[i][k])*(b[k][j]));
			}
		}
	}
	for(int i = 0; i < 2; i++){
		for(int j = 0; j < 2; j++){
			a[i][j] = res[i][j]%10;
		}
	}
}

void power(int (&a)[2][2], int n){
	if(n == 1) return;
	int m[2][2] = {{1,1},{1,0}};
	power(a,n/2);
	multiplicar(a,a);
	if(n%2 != 0) multiplicar(a,m);
}

//PROBLEM:
//LINK:
int main()
{
	ios::sync_with_stdio(0);
	cin.tie(0);
	#ifndef ONLINE_JUDGE
		freopen("_input.txt", "r", stdin);
		freopen("_output.txt", "w", stdout);
	#endif
	int tc; cin >> tc;
	while(tc--){
		int a[2][2] = {{1,1},{1,0}};
		int n; cin >> n;
		power(a,n-1);
		cout << (((a[0][0])%10)+(a[0][1]%10))%10 << endl;

	}
	return 0;
}
```

2.3 molly and chemicals

```
#include <bits/stdc++.h>
#define forn(i, n) for (int i = 0; i<(int)n; ++i)
#define forr(i, t, n) for (int i = t; i<n; ++i)
#define rmod(x, y) (((x%y)+y)%y)
using namespace std;
typedef unsigned int uint;
typedef long long ll;
typedef unsigned long long ull;
typedef pair<int, int> pii;
const double PI = acos(-1);
const int MAXN = 1e5+1;

int n, k;
ll a[MAXN];
ll dp[MAXN];
vector<ll> powers;
map<ll, ll> seen;

void genpowers(){
	if (k == 1) {powers.emp(k); return;}
	if (k == -1) {powers.emp(1); powers.emp(-1); return;}
	ll curp = 1;
	while(curp<=(1e14+100) and curp>=-(1e14 +100)) {
		powers.emp(curp);
		curp*=k;
	}
}
int main() {
	ios::sync_with_stdio(0);
```

```
        cin.tie(0);
        cin >> n >> k;
        genpowers();
        cin >> a[0];
        dp[0] = a[0];
        forr(i, 1, n) {
                cin >> a[i];
                dp[i] = dp[i-1]+a[i];
        }
        ll ans = 0, aux;
        forr(i, 0, n) {
                seen[dp[i]]++;
                for(auto&p:powers) {
                        aux = dp[i]-p;
                        if (aux == 0) ans++;
                        ans+=seen[aux];
                }
        }
        cout << ans;
        return 0;
}
```

3 Programación dinámica

3.1 Maximal Subrectangle

```
#include <bits/stdc++.h>
#define forn(i, n) for (int i = 0; i<(int)n; ++i)
#define forr(i, t, n) for (int i = t; i<n; ++i)
#define rmod(x, y) (((x%y)+y)%y)
using namespace std;
typedef unsigned int uint;
typedef long long ll;
typedef unsigned long long ull;
typedef pair<int, int> pii;
const int maxn = 100;

int dp[maxn][maxn];
int mat[maxn][maxn];
int arr[maxn];
int n;

int solve(){
        int tmax = mat[0][0];
        //column prefix sum
        forn(i, n) {
                forn(j, n)    {
                        if (not i) dp[i][j] = mat[i][j];
                        else dp[i][j]= dp[i-1][j]+mat[i][j];
                }
        }
        forn(start, n-1) {
                forr(end, start+1, n) {
                        forn(i, n) arr[i] = dp[end][i]-dp[start][i];
                        int aux = arr[0];
                        int tmx = aux;
                        forr(i, 1, n) {
                                aux = max(aux+arr[i], arr[i]);
                                tmx = max(tmx, aux);
                        }
                        tmax = max(tmax, tmx);
                }
        }
        return tmax;
}
int main() {
        ios_base::sync_with_stdio(0);
        cin.tie(0);
        cin >> n;
        forn(i, n)
                forn(j, n) cin >> mat[i][j];
        cout << solve();
        return 0;
}
```

3.2 Maximum Product Subarray

```
import sys
import math
sz = len
cout = print
ll = int
maxi = max
mini = min
rng = range
```

```
inf = float('inf')
for line in sys.stdin:
    a = [ll(j) for j in line.split() if ll(j) != (-999999)]
    n = sz(a)
    mxt = a[0]
    mx = a[0]
    mn = a[0]
    for i in rng(1, n):
        aux = mx
        mx = maxi(a[i], mx*a[i], mn*a[i])
        mn = mini(a[i], mn*a[i], aux*a[i])
        mxt = max(mxt, mx)
    print(mxt)
```

3.3 Set Cover using bitmask

```
#include <bits/stdc++.h>
#define forn(i, n) for (int i = 0; i<(int)n; ++i)
#define forr(i, t, n) for (int i = t; i<n; ++i)
#define rmod(x, y) (((x%y)+y)%y)
using namespace std;
typedef unsigned int uint;
typedef long long ll;
typedef unsigned long long ull;
typedef pair<int, int> pii;
const int maxn = 1000;
const int maxm = (1<<10);

int tc, n, d;
string aux;
int dp[maxn+1][maxm+1];
int masks[maxn];
int ans;
int mask(string& s) {
        int msk = 0;
        for(auto&i:s)
                msk |= (1<<(i-'0'));
        return msk;
}
bool complete(int x){
        return x+1 == (1<<10);
}
#define dpx dp[curn][curs]
const int decoy = (1<<30);
int solve(int curn, int curs)
{
        //dp[i][j]: minimo nmero de strings
        //requerido para formar la mascara j
        if (curn == n) {
                if (complete(curs)) return 0;
                else return decoy;
        }
        if (complete(curs)) return 0;
        if (dpx != -1) return dpx;
        //curs siempre sera menor o igual a 2^11-1;
        //2 opciones: uso el string actual o no
        //costo por paso: 1
        if ((curs|masks[curn])>curs)
                dpx = min(solve(curn+1, curs), 1 + solve(curn+1,
                    curs|masks[curn]));
        //si no hay mejora, por las puras se tomaria el string actual
        else dpx = solve(curn+1, curs);
        return dpx;
}
int main() {
        ios::sync_with_stdio(0);
        cin.tie(0);
        cin >> tc;
        while (tc--) {
                memset(dp, -1, sizeof(dp));
                cin >> n >> d;
                forn(i, n){
                        cin >> aux;
                        masks[i] = mask(aux);
                }
                ans = solve(0, 0);
                if (ans == decoy) cout << -1;
                else cout << ans;
                cout << "\n";
        }
        return 0;
}
```

3.4 Vacation

```
#include <bits/stdc++.h>
```

```
#define forn(i, n) for (int i = 0; i<(int)n; ++i)
#define forr(i, t, n) for (int i = t; i<n; ++i)
#define rmod(x, y) (((x%y)+y)%y)
using namespace std;
typedef unsigned int uint;
typedef long long ll;
typedef unsigned long long ull;
typedef pair<int, int> pii;
const int MAXN = 1E5;

int n;
int main() {
	ios::sync_with_stdio(0);
	cin.tie(0);
	cin >> n;
	vector<int> dp(3, 0);//previous day
	forn(d, n)	{
		vector<int> opts(3);
		forn(i, 3) cin >> opts[i];
		vector<int> ndp(3, 0);//new day
		forn(i, 3) {
			forn(j, 3) {
				if (i!=j)
					ndp[i] = max(ndp[i], dp[j]+opts[i]);
			}
		}
		dp = ndp;
	}
	cout << max({dp[0], dp[1], dp[2]});
	return 0;
}
```