jlcastrillon's blog

Counting problems from A to B

By jlcastrillon, 5 years ago,

Some problems ask to find how many numbers from A to B have a certain property, if the problem of finding how many numbers of k (0-9) digits have that property can be solved using a function F(k,property) in O(H) and when you update the left of a number the property can be updated in O(S) then there is a solution for the problem in O(H * S * log10^2(n)).

Let M(x) the amount of numbers less than x that have that property. Then M(B + 1) — M(A) is the solution to our problem, or M(B) — M(A) + h where (h = 1 if B have the property, else h = 0) To find M(x) we need to make a few observations. A number x less than y iff the length of x is less than the length of y or if they have equal length and there is a digit x[i] < y[i] and for all digits j < i this holds x[j] = y[j], that is ,they are equal from left to right until a digit i where x[i] < y[i], when this happens then all digits y[j] j > i can be in the range(0-9) and then F(k,property) can be used. We can use this to find all the numbers less than x having the desired property.

```
sol  = 0
remain = lengthof(x)
// we find the digit where they first differ x[i] < y[i] and for all
digits j < i x[j] = y[j]
for each digit x[i] of x from left to right{
    remain--;
    // now we find all the digits that can be at y[i] and are less than
x[i]
    for each digit d from 0 to x[i] - 1{
        property' = (property if digit d replaced digit x[i])
        sol += F(remain,property')
    }
    update property after deletion of digit x[i]
}
```

Here I have a sample C++ code to solve the following problem How many integers in the interval [A, B] are there such that the sum of their digits is S

```
#define ll long long
bool mem[N][N];
ll D[N][N];
// this is the function F(k,property)
ll F(int dig,int sum){
        if(dig == 0)
                return (ll)(sum  == 0);
        if(mem[dig][sum])
                return D[dig][sum];
        mem[dig][sum] = 1;
        ll ret = 0LL;
        for(int i = 0;i<=9;i++)
                ret += F(dig - 1,sum - i);
        D[dig][sum] = ret;
        return ret;
}

// this is M(x)
ll solve(ll x){
        ll ret = 0;
```

```
        sprintf(cad,"%lld",x);
        int len = strlen(cad);
        //sum is the desired property
        int sum = s;
        int qued = len;
        // we find the digit where they first differ x[i] < y[i] and for
all digits j < i x[j] = y[j]
        for(int i = 0;i < len;i++){
                qued--;
                int d = cad[i] - '0';
                // now we find all the digits that can be at y[i] and
are less than x[i]
                for(int j = 0;j <d;j++){
                        //sum - j = property'
                        if(sum - j >= 0){
                                ret += F(qued,sum - j);
                        }
                }
                //update property after deletion of digit x[i]
                sum -= d;
        }
        return ret;
}


//this is the solution to the problem
sol = solve(b + 1) - solve(a);
```

Some problems to solve

- http://www.spoj.com/problems/LUCIFER/
- http://www.spoj.com/problems/RAONE/
- http://www.spoj.com/problems/GONE/
- http://coj.uci.cu/24h/problem.xhtml?abb=2481
- http://coj.uci.cu/24h/problem.xhtml?abb=1242

and many other you can find anywhere

counting

+9 jlcastrillon 5 years ago 42

CountZero → Turn your C++/Java/whatever source into executable

HOMIARA_RUBY → About Java that I want to know

BledDest → Educational Codeforces Round 37 - Editorial

csacademy → Round #68 (Div. 2)

MemorySlices → Invitation to CodeChef February Long Challenge 2018!

ckd → How to solve this??

HARRYPOTTER0 → How to prepare for ACM ICPC?

sinhashubham95 → Invitation to February Easy'18 on Hackerearth

Unlimited_Time → Notes on Codeforces Beta Round #115, Div2- A, B, C, D

shash42 → Approach to 2D Segment Tree with Lazy Propogation (should allow RMQ?)

meret → Editorial for Codeforces Round #134

fcspartakm → Codeforces Round #452 (Div.2) Editorial

legendjohn9999 → How to find the diameter of a graph, which have only one cycle(N vertexes, N paths)?

Unlimited_Time → Notes on Codeforces Beta Round #114, Div2- A, B, C, D, E (a delicately designed mathematical problem)

giorgosgiapis → COCI 2017/2018 — Round 6

craus → Codeforces Round #335 Problem Analysis

Valour → Palindromic Queries

jinlifu1999 → Codeforces Round #460 (Div. 2) Editorial

LelouchRiBritannia → 843E AIM Tech Round 4 (Div. 1) tourist's solution

Harolinch → alternating subsequance problem

gKseni → Design competition for Codeforces T-shirts

Harolinch → [blog]working on queries

Detailed →

Comments (42)

Write comment?

ylguirola

5 years ago, # | +1

it is good

→ Reply

jlcastrillon

5 years ago, # | ← Rev. 6 0

these are other similar problems http://lightoj.com/volume_showproblem.php?problem=1205 http://www.spoj.com/problems/NUMTSN/

→ Reply

abhishekg

4 years ago, # ^ | 0

@jlcastrillon can you please check my code what's wrong in it for http://www.spoj.com/problems/NUMTSN/ problem.... it is giving TLE

→ Reply

4 years ago, # ^ | ← Rev. 6 0

abhishekg

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

using namespace std;

long long int mod=1000000007;

long long int d[51][51][51][51];

bool mem[51][51][51][51];

int len;

long long int f(int i, int three, int six, int nine, int lo, char *cad) {

if (i == len)
{

    if(three==six && six==nine)
                    return 1;
            else
                    return 0;
}


long long int ret = 0;

int dig;


if (lo) {
    if (mem[i][three][six][nine]) {
       return d[i][three][six][nine];
    } else {
       mem[i][three][six][nine] = 1;
       long long int r = 0;
       for (dig = 0; dig <= 9; ++dig) {
         if(dig==3)
                                    r=(r+f(i +
1,three+1, six,nine, lo, cad))%mod;
                            else if(dig==6)
                                    r=(r+f(i + 1,three,
six+1,nine, lo, cad))%mod;
                            else if(dig==9)
                                    r=(r+f(i + 1,three,
six,nine+1 , lo, cad))%mod;
                            else
                                    r=(r+f(i + 1,three,
six,nine, lo, cad))%mod;
       }
       d[i][three][six][nine] = r%mod;
       return r%mod;
    }
}

int limit;
limit = cad[i] - '0';

for (dig = 0; dig <= limit; ++dig) {
                            if(dig==3)
                                    ret=(ret+f(i +
1,three+1, six,nine, (lo || (dig < (cad[i] - '0'))),
cad))%mod;
                            else if(dig==6)
                                    ret=(ret+f(i +
1,three, six+1,nine, (lo || (dig < (cad[i] - '0'))),
cad))%mod;
                            else if(dig==9)
                                    ret=(ret+f(i +
1,three, six,nine+1 , (lo || (dig < (cad[i] - '0')))
```

```
1,three, six,nine+1 , (lo || (dig < (cad[i] - '0'))),
cad))%mod;
                                else
                                        ret=(ret+f(i +
1,three, six,nine, (lo || (dig < (cad[i] - '0'))),
cad))%mod;
}
return ret%mod;

}

long long int solve(char *x) { len = strlen(x);

memset(d, 0, sizeof(d));

memset(mem, 0, sizeof(mem));

return f(0, 0, 0,0, 0, x);

}

char aa[51];

char bb[51];

int check(char *x) { int a=0,b=0,c=0,i,j,k;

k=strlen(x);

    for(i=0;i<k;i++){

            if(x[i]=='3')
                    a++;

            else if(x[i]=='6')
                    b++;

            else if(x[i]=='9')
                    c++;

    }

    if(a==b && b==c)
            return 1;
    else
            return 0;

}

int main() { int t;

long long int sol;

     char r;

    scanf("%d",&t);

    while(t--){

            scanf("%s",&aa);

            scanf("%c",&r);

            scanf("%s",&bb);

            scanf("%c",&r);

            sol = (solve(bb) - solve(aa))%mod;

            sol= sol + check(aa);
```

```
                printf("%lld\n",sol%mod);

        }
        return 0;

}
```

→ Reply

abhishekg

4 years ago, # ^ | ← Rev. 4 0

accepted now :)

→ Reply

4 years ago, # ^ | ← Rev. 2 0

I'm trying to solve the same problem and getting TLE. What did you improve in your code? I don't know what else to do :s

This is the main function of my code. Could you please help me?

```
int f(int i, int tres, int seis, int nueve, bool menor)
{
        int piv = max(tres, max(seis, nueve));

        if ( piv-nueve + piv-seis + piv-tres > n-i or (piv == 0 and n-i < 3) )
                return 0;

        if (i == n)
                return tres == seis and tres == nueve and tres > 0;

        if (dp[i][tres][seis][nueve][menor] != -1)
                return dp[i][tres][seis][nueve][menor];

        int res = 0, end = menor ? 9 : x[i] - '0';
        For(d, 0, end+1)
                res = ( res + f( i+1, tres + (d == 3), seis + (d == 6), nueve + (d == 9), menor | (d < x[i] - '0') ) ) % MOD;

        return dp[i][tres][seis][nueve][menor] = res;
}
```

xdavewk

→ Reply

ABHISHEK004

4 years ago, # ^ | ← Rev. 2 0

i also have the same problem

→ Reply

saurabh060792

3 years ago, # ^ | 0

You don't need 5 dimensional dp(it had given me tle when I used 4D dp). Try solving it by combinatorics.

→ Reply

mjnovice

3 years ago, # ^ | 0

I implemented in a very similar manner, however am getting wrong answer.

```
long long solve(int i, int a3, int a6, int
```

```
long long solve(int i,int a3,int a6,int
a9,int lo)
{
    int n = d.length();
    if(i==n )
        return (a3==a6) && (a6==a9) &&
(a3>=1);
    if(a3>17 || a6>17 || a9>17)
        return 0;


    if(dp[i][a3][a6][a9][lo]!=-1)
        return dp[i][a3][a6][a9][lo];

    dp[i][a3][a6][a9][lo]=0;
    for(int digit=0;digit<=(lo?9:(d[i]-
'0'));digit++)
    {
        long long tmp=solve(i+1,a3+
(digit==3),a6+(digit==6),a9+
(digit==9),lo||(d[i]-'0')>digit);
        assert(tmp>=0 && tmp<MOD);
        dp[i][a3][a6][a9][lo] = (dp[i]
[a3][a6][a9][lo] + tmp)%MOD;
    }
    assert(dp[i][a3][a6][a9][lo]>=0);
    return dp[i][a3][a6][a9][lo];
}
```

→ Reply

mjnovice

3 years ago, # ^ | 0

solved! :) AC! Thanks a lot people for the valuable insights.

→ Reply

pk845

3 weeks ago, # ^ | 0

can you please share your accepted code?

→ Reply

HandleNeeded

4 years ago, # ^ | 0

Google Code jam 2014 Round1 B Problem B is a good problem of this kind.^_^

→ Reply

niklasb

4 years ago, # ^ | 0

Indeed, I wrote about the solution here

→ Reply

r-neo

4 years ago, # | 0

Hey nice article , can you please give link to some working code of the problem . like I have lot of trouble writing the F(k, property) in different cases...

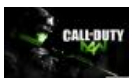
→ Reply

chrome

4 years ago, # ^ | +3

For example: Solution for Last contest 431D

→ Reply



r-neo

4 years ago, # | 0

thanks for the reply , but in the above case the sum (s) is given , so we are able to get the difference and calculate it, but in some cases , like where 1. some

property of the difference b/w the sum of the even place digits and odd place

property of the difference b/w the sum of the even place digits and odd place digits must have some property like being prime number or diff should be 1 .

Is there any tutorial which tells how to formulate 3-Dimensional dp for it.

→ Reply

vipul.jain

4 years ago, # ^ | +2

:)

```
int d[10][50][50];
bool mem[10][50][50];
int len;

int f(int i, int sum_odd, int sum_even, int lo, const
string& cad)
{
        if (i == len) {
                return (sum_even - sum_odd == 1);
        }
        int ret = 0;
        int dig;

        if (lo) {
                if (mem[i][sum_odd][sum_even]) {
                        return d[i][sum_odd][sum_even];
                } else {
                        mem[i][sum_odd][sum_even] = 1;
                        int r = 0;
                        for (dig = 0; dig <= 9; ++dig)
{
                                if ((len - i) % 2 == 0)
{
                                        r += f(i + 1,
sum_odd, sum_even + dig, lo, cad);
                                } else {
                                        r += f(i + 1,
sum_odd + dig, sum_even, lo, cad);
                                }
                        }
                        d[i][sum_odd][sum_even] = r;
                        return r;
                }
        }

        int limit;
        limit = cad[i] - '0';

        for (dig = 0; dig <= limit; ++dig) {
                if ((len - i) % 2 == 0) {
                        ret += f(i + 1, sum_odd,
sum_even + dig, (lo || (dig < (cad[i] - '0'))), cad);
                } else {
                        ret += f(i + 1, sum_odd + dig,
sum_even, (lo || (dig < (cad[i] - '0'))), cad);
                }
        }
        return ret;
}

int solve (int x)
{
        string cad;
        cad = NumtoString(x);
        len = cad.length();
        memset(d, 0, sizeof(d));
        memset(mem, 0, sizeof(mem));

        return f(0, 0, 0, 0, cad);
```

```
return f(0, 0, 0, 0, cad);
}
```

→ Reply

vipul.jain

4 years ago, # ^ | +3

my implementation is guided by this post

http://stackoverflow.com/questions/22394257/how-to-count-integers-between-large-a-and-b-with-a-certain-property/22394258#22394258

:)

→ Reply

flickr

4 years ago, # | 0

How it is covering all the numbers less than 123 (say)? first we choose 1 and varied it from 0- 0 -> it covers 000- 099 then we choose 2 and varied it from 0- 1 -> it covers 000- 019. please give some explanation!

→ Reply

jlcastrillon

4 years ago, # ^ | 0

for 123 when changing 1 it covers from 000 to 099 when changing 2 it covers from 100 to 119 when changing 3 it covers from 120 to 122

to calculate how many numbers less than X have certain property iterate through all possible positions where a number Y may differ for the first time when compared with X and through all possible digits for that position. you can easily notice that the rest(all the digits to the right of that position) may take values from 0 to 9, then then if you have a function(almost always solvable by dp) to calculate how many numbers with n(any amount) of digits have certain property(for example a sum equal to S) then your problem is solved.

→ Reply

flickr

4 years ago, # ^ | ← Rev. 3 0

thanks, awesome tutorial!

→ Reply

ABHISHEK004

4 years ago, # | 0

@jlcastrillon nice article...

but can you please tell me how to find count of numbers between a and b which contains 0 as their digit... i am not able to get this with above idea, it becomes very complex in case of leading zeroes

please give some explanation as i am having lot of trouble with this ...

waiting for ur reply

→ Reply

jlcastrillon

4 years ago, # ^ | 0

first of all find a dp solution to the problem when all digits can be form 0 to 9 and having a fixed number of digits. Then when calculating for a number X how many numbers are less than it and cantain at least one zero, don't take into account the zero digit when changing the value of the first digit this will give you as a solution all the numbers that contain at least one zero with the same number of digits than X and dont't contain leading zeros, then add to the solution how many numbers with less digits than X are there that contain at least one zero and don't contain leading zeroes. Have in mind that you need a special case dp that tells you the solution when the first digit cannot be zero for the second part of the solution.

→ Reply

HandleNeeded

4 years ago, # | ← Rev. 2 0

I write a blog on my website discussing the skill to solve problems of this kind with a contest I created consisting of almost every problem mentioned in this blog or comment. It's a pity that I wrote it in Chinese. So if you are interested in it and you can read Chinese, CLICK!

→ Reply

xdavewk

4 years ago, # ^ | 0

Thank you very much. I cant read Chinese, but i was able to find the code of the problem i was getting TLE and learned about using a - b instead of a and b and check for a - $b = 0$ instead of $a = b$.

→ Reply

HandleNeeded

4 years ago, # ^ | 0

You're welcome.:D

→ Reply

I_Love_Bofei

2 years ago, # ^ | 0

YQCMMD, plz update the link to http://yqcmmd.com/2014/06/12/%E6%95%B0%E4%BD%8Ddp%E4%B8%93%E9%A2%98/ in case of future reference.D

Thanks a lot for this article!

→ Reply

HandleNeeded

2 years ago, # ^ | 0

Done.

→ Reply

trainsust

4 years ago, # | 0

what RET means in English?Can you explain other short words usually used? Thanks!

→ Reply

HandleNeeded

4 years ago, # ^ | 0

return for short

→ Reply

Vital1ty

4 years ago, # | 0

Hi....:)

I am working on the RAONE problem on spoj

link:http://www.spoj.com/problems/RAONE/

I follwed the same way....but am not getting the right answer.....

heres the link to my answer:

http://ideone.com/5CemTn

pls...tell me where i am goin wrong

thnx in advance :)

→ Reply

4 years ago, # ^ | +3

did mistake on the parity check :)

have got it accepted now

Vital1ty

have got it accepted now

thnx for this wonderful tutorial :)

→ Reply

anmol_varshney

3 years ago, # ^ | 0

I am trying to solve the same problem i.e. RAONE on SPOJ, but i can't grasp the idea behind it. Can you please explain me the approach of the same in respect to the above blog. It would be very useful :D

→ Reply

mesayantan

3 years ago, # | -14

I am trying to solve the problem http://www.spoj.com/problems/NUMTSN/ I am getting WA. Help me finding the bug. Here is my code.

include
include
include
include
include
include
include
include
include
include
define s(a) scanf("%d",&a)
define sc(a) scanf("%s",&a)
define p(a) printf("%d\n",a)
define pf(a) printf("%lld\n",a)
define f(i,r) for(int i=0;i<r;i++)
define fr(i,p,r) for(int i=p;i<r;i++)
define ll long long
define mod 1000000007

using namespace std;

char num[55];

ll dp[55][55][55][55];

int len;

ll cal(int dig,int a,int b,int c) {

```
if(dig==0)
return ((a==b) && (b==c) && (a>0));

if(dp[dig][a][b][c]!=-1)
return dp[dig][a][b][c];

int s1,s2,s3;
```

```
int s1,s2,s3;
ll count=0;

f(j,10)
{
    s1=s2=s3=0;

    if(j==3)
    s1++;
    else if(j==6)
    s2++;
    else if(j==9)
    s3++;

    count+=cal(dig-1,a+s1,b+s2,c+s3);

    if(count>=mod)
    count%=mod;
}

dp[dig][a][b][c]=count;

return count;

}

ll solve() {

int dig=len;
int s1,s2,s3,a=0,b=0,c=0;
ll count=0;

f(i,len)
{
    dig--;
    int d=num[i]-'0';

    f(j,d)
    {
       s1=s2=s3=0;

       if(j==3)
       s1++;
       else if(j==6)
       s2++;
       else if(j==9)
       s3++;

       count+=cal(dig,a+s1,b+s2,c+s3);

       if(count>=mod)
       count%=mod;

    }

    if(d==3)
    a++;
    else if(d==6)
    b++;
    else if(d==9)
    c++;
}

return count;

}

int main() {
int t;
```

```
int t;
ll x,y;

memset(dp,-1,sizeof(dp)/sizeof(dp[0][0][0][0]));

s(t);
while(t--)
{
    sc(num);
    len=strlen(num);

    int i;

    for(i=len-1;i>=0;i--)
    {
       if(num[i]=='9')
       num[i]='0';
       else
       {
         num[i]++;
         break;
       }
    }

    if(i<0)
    {
       for(int j=len-1;j>=0;j--)
       num[j+1]=num[j];

       num[0]='1';
    }

    //cout<<"num "<<len<<endl;

    x=solve();

    sc(num);
    len=strlen(num);


    for(i=len-1;i>=0;i--)
    {
       if(num[i]=='9')
       num[i]='0';
       else
       {
         num[i]++;
         break;
       }
    }

    if(i<0)
    {
       for(int j=len-1;j>=0;j--)
       num[j+1]=num[j];

       num[0]='1';
    }

    //cout<<"num "<<num<<endl;

    y=solve();

    x=y-x;

    if(x<0)
    x+=mod;
```

```
        pf(x%mod);

    }

    }
```

→ Reply

mesayantan

3 years ago, # ^ | ← Rev. 2 0

ACed.Mistook absent mindedly.1st string shouldnot be increased by 1.sorry.

→ Reply

allrounder

3 years ago, # | ← Rev. 3 0

I an getting WA could not figure out what the provlem is my code is

include<bits/stdc++.h>

using namespace std;

define gc getchar//_unlocked
define pc putchar//_unlocked
define pb push_back
define mp make_pair-
define f first
define s second
define MAXN 100005
define MOD 1000000007
define mod(a,b) a>b?a-b:b-a
define ll long long
define pii pair< ll,ll >

using namespace std;

inline void inp(ll *n) { *n=0; ll ch=gc(); int sign=1; while(ch < '0' || ch > '9') {if(ch=='-')sign=-1; ch=gc();} while(ch >= '0' && ch <= '9') *n = (*n<<3)+(*n<<1) + ch-'0', ch=gc(); *n=*n*sign; }

inline void fastp(ll a) { register char c; char snum[20]; int i=0; do { snum[i++]=a%10+48; a=a/10; }while(a!=0); i=i-1; while(i>=0) pc(snum[i--]); pc('\n'); }

define N 52

bool mem[N][N][N][N];

ll F(ll dig,ll threes,ll sixes,ll nines) { if(!dig) return (ll)(threes&&(threes==sixes)&& (sixes==nines));

```
if(mem[dig][threes][sixes][nines]!=-1) return mem[dig]
[threes][sixes][nines];

ll ret = 0LL;
ll a,b,c;
for(ll i=0;i<=9;i++)
{
    a=b=c=0;
    if(i==3) a++; if(i==6) b++; if(i==9) c++;
```

```
    if(i==3) a++; if(i==6) b++; if(i==9) c++;
    ret += (F(dig - 1,threes+a,sixes+b,nines+c)%MOD);
    ret%=MOD;
}
mem[dig][threes][sixes][nines] = ret;
return ret;

}
```

ll solve(char s[],ll len) { ll ret = 0; ll threes,sixes,nines,sum; sum=threes=sixes=nines=0; ll qued = len; for(ll i = 0;i < len;i++) { qued--; ll d=s[i]-'0'; ll a,b,c; ll temp=0LL; for(ll j=0;j<d;j++) { a=b=c=0; if(j==3) a++; if(j==6) b++; if(j==9) c++; ret+=F(qued,threes+a,sixes+b,nines+c)%MOD; ret%=MOD; } if(d==3) threes++; if(d==6) sixes++; if(d==9) nines++; } return ret; }

int main() { ll t,n,i,j,l,r,m,k; //freopen("x.txt","r",stdin); char a[52],b[52]; memset(mem,-1,sizeof(mem)); inp(&t); while(t--) { scanf("%s",a); scanf("%s",b); ll ans=0; l=strlen(a); r=strlen(b); ll x=0,y=0,z=0; for(i=0;i<r;i++) { if(b[i]-'0'==3) x++; if(b[i]-'0'==6) y++; if(b[i]-'0'==9) z++; } if(x&&(x==y)&&(y==z)) ans++; ans+=solve(b,r)%MOD; ans%=MOD; ans=(ans — solve(a,l)+MOD)%MOD; printf("%lld\n",ans%MOD); } return 0; } ~~~~~

```
Your code here...
```

~~~~~

→ Reply

allrounder

3 years ago, # ^ | 0

Got AC ....was doing a silly mistake

→ Reply

xplorer

3 years ago, # | 0

Can anyone explain how to solve this on codechef using above approach ? Any kind of help will be appreciated. Thanks in advance.

→ Reply

asif_1152

3 years ago, # | 0

awesome explanation, thanks

→ Reply

ACrush_ujn

2 years ago, # | 0

A good read but can anyone make the recursive tree of the problem which led to DP solution!

→ Reply

Tomah4wk

2 years ago, # | 0

http://www.spoj.com/problems/LOTGAME/

I recently added this one to SPOJ :D

→ Reply

prathmesh_01

5 months ago, # | 0

I am trying to solve http://www.spoj.com/problems/RAONE/ but getting TLE can anyone please tell me whats wrong ,thanks;) ~~~~~ import java.util.*;

class hello { static long[] y; static long[][][] dp; static int n; static int conv(long mid) { int sum=0; long temp=mid; while(temp!=0) { sum+=temp%10; temp/=10; } return sum; } hello(long b) {

```
String g=String.valueOf(b);
    n=g.length();

    y= new long[g.length()+1];
    long f=b;
    int i=n-1;
```
~~~~~

```
    int f=n-1;
    dp= new long[20][100][2];
    while(f!=0)
    {
        y[i--]=f%10;
        f=f/10;
    }
    for( i=0;i<20;i++)for(int j=0;j<100;j++){dp[i][j]
[0]=dp[i][j][1]=-1;}

}

// esum= even sum ,osum= oddsum
static long f(int index ,int esum, int osum ,int r)
{
    if(index>=n)
    {
      //BASE CASE
        return ( esum-osum==1?1:0)  ;
    }
    if(esum>=osum)
    if(dp[index][esum-osum][r]!=-1) return dp[index][esum-
osum][r];

    long ans=0;

    if(r==0)
    {
       for(int i=0;i<=9;i++)
         ans+=f(index+1 ,esum+(((n-index)%2==0)?i:0) ,osum+
(((n-index)%2==1)?i:0),r);
    }
    else
    {
       for(int i=0;i<=9&&i<=y[index];i++)
       {
         if(i<y[index])
         ans+=f(index+1 ,esum+(((n-index)%2==0)?i:0),osum+
(((n-index)%2==1)?i:0),0);
         else if(i<=y[index])
         ans+=f(index+1,esum+(((n-index)%2==0)?i:0), osum+
(((n-index)%2==1)?i:0),1);
         else break;
       }

    }
    if(esum-osum>=0)
    dp[index][esum-osum][r]=ans;
return ans;
}

public static void main(String args[] ) throws Exception {
    Scanner s = new Scanner(System.in);
    int t= s.nextInt();
   while(t-->0){

   long a = s.nextLong();
   long b = s.nextLong();

//Solve for b
    hello test= new hello(b);
    long a1=f(0,0,0,1);

    //Solve for a-1
   hello tes= new hello(a-1);
    long a2=f(0,0,0,1);

    System.out.println((a1-a2));
   }
 }

~~~~~
```

ʃ

→ Reply
