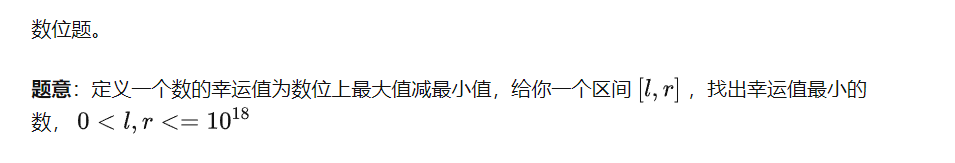
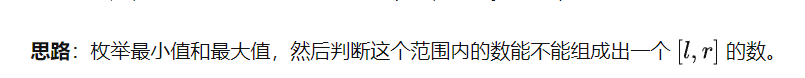
https://codeforces.com/contest/1808/problem/C



当然有一种比较简单的方法,可以直接



不过也有比较有价值的解法,可以用数位dp来做

#define mpa make\_pair

typedef long long LL;

typedef long double LD;

typedef pair<int, LL> PII;

typedef pair<db, db> PDD;

typedef vector<int> vci;

constexpr int N = 1e5 + 10, M = 2e6 + 10, B = 21, md = 1e9 + 7;

int T, n, m;

PII s = mpa(-1, -1);

PII dp[B][10][10];

LL e[B];

void init()

{

e[0] = 1;

lop(i, 1, B)

e[i] = e[i - 1] \* 10;

lop(i, 0, B)

rep(j, 0, 9)

rep(k, 0, 9)

dp[i][j][k] = s;

}

int a[B], b[B];

PII dfs(int pos, bool limit1, bool limit2, bool lead0, int mx, int mi)

{

if(!pos)

{

return mpa(mx - mi, 0);

}

auto &now = dp[pos][mx][mi];

if(!limit1 && !limit2 && !lead0 && now != s)

{

return now;

}

int down = limit1 ? a[pos] : 0;

int up = limit2 ? b[pos] : 9;

int res = 10;

LL tmp;

rep(i, down, up)

{

bool new0 = lead0 && i == 0;

int mmx = new0 ? 0 : max(mx, i);

int mmi = new0 ? 9 : min(mi, i);

auto [u, v] = dfs(pos - 1, limit1 && i == down, limit2 && i == up, new0, mmx, mmi);

if(res > u)

{

res = u;

tmp = i \* e[pos - 1] + v;

}

}

if(!limit1 && !limit2 && !lead0)

{

now = mpa(res, tmp);

}

return mpa(res, tmp);

}

LL solve(){

LL l, r;

cin >> l >> r;

int tmp = 0;

while(r > 0)

{

b[ ++ tmp] = r % 10;

r /= 10;

}

int len = tmp;

rep(i, 1, len)

a[i] = 0;

tmp = 0;

while(l > 0)

{

a[ ++ tmp] = l % 10;

l /= 10;

}

return dfs(len, true, true, true, 0, 9).second;

}

int main()

{

init();

cin.tie(0)->sync\_with\_stdio(false);

cin >> T;

while (T--)

cout << solve() << el;

}