

# 2020.07.18-2020.07.24 周报

## 团队训练

2020.07.18 [2020牛客暑期多校训练营（第三场）](#) prob:7:8:12 rnk:112/1175

[20200718比赛记录](#)

2020.07.20 [2020牛客暑期多校训练营（第四场）](#) prob:4:5:10 rnk:60/1112

[20200720比赛记录](#)

2020.07.23 [2020.07.23codeforces加训](#) prob:6:6:11 rnk:7/17

[20200723比赛记录](#)

## \_wzx27

### 专题

做了点后缀自动机的题。

### 题目

牛客三

[A - Clam and Fish](#) [F - Fraction Construction Problem](#)

牛客四

[F - Finding the Order](#) [I - Investigating Legions](#)

### 比赛

[Atcoder Beginner Contest 127 \(VP\)](#)

[Atcoder Beginner Contest 128 \(VP\)](#)

## Infinity37

### 专题

无

## 题目

牛客三

[B - Classical String Problem](#) [D - Points Construction Problem](#)

牛客四

[B - Basic Gcd Problem](#) [C - Count New String](#)

## 比赛

[cfr659 div2\\_infinity37](#)比赛记录

## Zars19

### 专题

[20200719 - 一些非常简单的计算几何扫描线题](#)

## 题目

### 比赛

[Codeforces Round 658 \(Div. 2\) Zars19](#)

## 本周推荐

### [\\_wzx27](#)

题目链接：[Takahashi's Basics in Education and Learning](#)

tag [矩阵快速幂](#) [思维](#)

题意：

给一个等差数列的首项  $A$  公差  $B$  和项数  $L$ 把这  $L$  项连接起来得到一个很长的整数。求这个数对  $M$  取模的结果。

$1 \leq L, A, B \leq 10^{18}$

$1 \leq M \leq 10^9$

保证等差数列的每一项都不超过  $10^{18}$ 。

题解：

如果暴力维护答案就是  $ans = ans \times 10^{\text{bits}} + a_i$  这种线性递推式考虑用矩阵快速幂优化。

把  $L$  项按照位数划分，对于每一组位数为  $\text{bits}$  的项，都构造如下矩阵：

$$\begin{bmatrix} 10^{\text{bits}} & 1 & 0 \\ 0 & 1 & B \\ 0 & 0 & 1 \end{bmatrix}$$

左乘矩阵

$$\begin{bmatrix} ans \\ a_i \\ 1 \end{bmatrix}$$

代码：

```
#include<bits/stdc++.h>
#define ll long long
#define pii_ pair<int,int>
#define mp_ make_pair
#define pb push_back
#define fi first
#define se second
#define rep(i,a,b) for(int i=(a);i<=(b);i++)
#define per(i,a,b) for(int i=(a);i>=(b);i--)
#define show1(a) cout<<#a<<" = "<<a<<endl
#define show2(a,b) cout<<#a<<" = "<<a<<" "; cout<<#b<<" = "<<b<<endl
using namespace std;
const ll INF = 1LL<<60;
const int inf = 1<<30;
const int maxn = 2e5+5;
inline void fastio() {ios::sync_with_stdio(false);cin.tie(0);cout.tie(0);}
ll qpow(ll a,ll b,ll M) {a%=M;ll
s=1;while(b){if(b&1)s=(s*a)%M;a=(a*a)%M;b>>=1;}return s; }
ll qmul(ll a,ll b,ll M) {a%=M;ll
s=0;while(b){if(b&1)s=(s+a)%M;a=(a+a)%M;b>>=1;}return s; }
ll M,pt[20];
struct Matrix
{
    ll mat[3][3];
    Matrix() {memset(mat,0,sizeof(mat));}
    Matrix operator * (Matrix b)
    {
        Matrix res;
        rep(i,0,2) rep(j,0,2) rep(k,0,2) res.mat[i][j] = (res.mat[i][j] +
mat[i][k]*b.mat[k][j]%M)%M;
        return res;
    }
    Matrix operator ^ (ll b)
    {
        Matrix res,A=*this;
        rep(i,0,2) res.mat[i][i] = 1;
        while(b){
```

```
        if(b&1) res = res * A;
        A = A*A;
        b>>=1;
    }
    return res;
}
};
int main()
{
    fastio(); ll n,a,b;
    cin>>n>>a>>b>>M;
    ll ans = 0;
    pt[0] = 1;
    rep(i,1,18) pt[i] = pt[i-1]*10;
    ll L = a,R = a+(n-1)*b;
    rep(i,1,18){
        if(L < pt[i]){
            ll e = (pt[i]-L-1)/b*b+L;
            if(e>R) e = R;
            ll t = (e-L)/b + 1;
            Matrix A;
            A.mat[0][0] = pt[i]%M;
            A.mat[0][1] = A.mat[1][1] = A.mat[2][2] = 1;
            A.mat[1][2] = b%M;
            A = A^t;
            ans = (ans*A.mat[0][0]%M + L%M*A.mat[0][1]%M + A.mat[0][2]) %M;
            if(e==R)break;
            L = e + b;
        }
    }
    cout<<ans<<endl;
    return 0;
}
```

推荐理由：

做矩阵快速幂的题只在刷专题的时候做过，然而比赛的时候容易想不到，可以用来提示一下^。以及这个按位拆分也挺有意思的（吧）。

## Zars19

来源：在做POJ 3004 Subway Planning的时候看到了这个——[三类贪心区间覆盖问题 BY Ra煞](#)

tag：贪心。

### 概述

1. 给出  $n$  个区间，问最少使用多少个可以实现某个大区间的完全覆盖？

2. 给出  $n$  个区间，问最多选择多少个可以使得两两不重叠？
3. 给出  $n$  个区间，问最少选择多少点可以使得每个区间中都有至少一个点（点可被共用）？

### 答案

1. 排序，枚举起始区间，之后选择左端点在当前覆盖范围内且右端点最大的一个，重复直至目标达成。
2. 排序（第一关键字左端点从大到小，第二关键字右端点从小到大），按顺序尝试加入每一个区间（这个顺序可以维持留给左边的空白区域达到极大），如果重叠则丢掉。
3. 排序（第一关键字左端点从小到大，第二关键字右端点从小到大），按顺序加入，维护当前重叠区域的右端点，如果新区间左端点大于当前右端点，则丢掉之前维护的区域答案++增加一个点，否则如果新区间右端点小于当前右端点，更新右端点。

**comments** : 虽然不难但可能会在出现在奇怪的辅助位？可以撕烤一下。

From:

<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:

[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:wangzai\\_milk:weekly12&rev=1595574704](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:wangzai_milk:weekly12&rev=1595574704)

Last update: **2020/07/24 15:11**