

Solution 7.26

T1 ball

先去除前导零，然后一个简单的高精度对低精度除法

【核心代码】

```
cin >> s;
for (int i = 0; i < s.size(); i++) {
    if (s[i] != '0') {
        p = i;
        break;
    }
}
for (int i = p; i < s.size(); i++) {
    ans *= 10;
    ans += s[i] - '0';
    if (ans >= 7) {
        ans %= 7;
    }
}
```

T2: number

N^2 的后九位是 M

方法一：按位搜索，从后往前搜索，每多放一位就平方一下计算后几位是否一致，比如 M 的最后三位只收到 N 的最后三位的影响。

方法二：折半查找，先枚举满足要求的后 5 位，把满足要求的后五位找出来，再枚举前 4 位进行拼接从而找出符合要求的数。

T3: Cut

标签：贪心

本题为贪心题，我们按照横切 和 竖切 分别排序。

如果现在选择横切 $h[i]$ ，那么假设竖切已经进行了 j 下，那么代价就是 $h[i] * j$

如果现在选择竖切 $s[j]$ ，那么假设横切已经进行了 i 下，那么代价就是 $s[j] * i$

因此可以是一个类似于归并排序的贪心即可。

【核心代码】

```
sort(dh + 1, dh + n, cmp);
sort(ds + 1, ds + m, cmp);
long long i = 1, j = 1;
while (i < n && j < m) {
    if (dh[i] >= ds[j]) {
        ans += dh[i] * j; ++i;
    } else {
        ans += ds[j] * i; ++j;
    }
}

while (i < n) {
    ans += dh[i] * m; ++i;
}
while (j < m) {
    ans += ds[j] * n; ++j;
}
```

T4 seq

回文子序列问题，我们可以进行区间 DP。

$f[l][r]$ 表示 从 $l \sim r$ 这个区间内容最长的回文子序列长度

$f[l][r] = \max(f[l][r - 1], f[l + 1][r], f[l + 1][r - 1] + 2)$ 如果第 l 个和第 r 个能够匹配);

区间 DP 可以考虑使用 记忆化搜索

判断两个数能否匹配，即 x 能否变换到 y 我们可以使用并查集或者 `dfs` 判断他们是否在一个连通块。

【核心代码】

```
int dfs(int l, int r) {
    if (f[l][r] != -1)
        return f[l][r];
    if (l > r)
        return f[l][r] = 0;
    if (l == r)
        return f[l][r] = 1;
    int ans = 0;
    if (a[l] == a[r])
        return dfs(l + 1, r - 1) + 2;
    ans = max(ans, dfs(l, r - 1));
    ans = max(ans, dfs(l + 1, r));
    return f[l][r] = ans;
}
```