

# Solution 7.29

## T1 f 函数

签到题，暴力模拟，直接写个递归函数也能 AC，因为递归不了几次。也可以记忆化一下。仔细观察，或者打表，对于小于等于 100 的 x 函数值 91，其余情况为  $x - 10$

## T2 异或和

40% 可以暴力获得

100% 因为 xor 运算每位是独立的，并且只有一个数该位上是 0，另一个数该位上是 1，才能产生贡献。因此我们可以逐一考虑每一位，加入当前考虑从右往左数的第 j 位（从第 0 位开始），n 个数字中该位上是 0 的有 x 个，该位上是 1 的有  $n - x$  个，那么该位对于总答案的贡献为：  $(1 \ll j) * (n - x) * x$

## T3 公约数

20% 可以暴力计算两个数然后求 gcd

100% 考虑对 A 和 B 分别进行质因子分解

假设  $A = p_1^{c_1} \times p_2^{c_2} \dots \times p_k^{c_k}$ ,  $B = p_1^{d_1} \times p_2^{d_2} \dots \times p_k^{d_k}$

则  $\gcd = p_1^{c_1, d_1} \times p_2^{c_2, d_2} \dots \times p_k^{c_k, d_k}$

## T4 最长公共上升子序列

20% 可以暴力搜索

50% 可以写一个  $O(n^3)$  的 dp （ $N^4$  太暴力）

$F[i][j]$  前 A 序列里面 前 i 个数中最后一个是与  $b[j]$  匹配的情况下的 最长公共上升子序列长度。

If  $a[i] == b[j]$      $f[i][j] = f[i - 1][k] + 1$     if ( $b[k] < b[j]$ )

If  $a[i] != b[j]$      $f[i][j] = f[i - 1][j]$

100% 考虑优化到 因为当前的  $b[j]$  要跟  $a[i]$  匹配，所以我们的  $b[k] < a[i]$  就能用来转移到  $f[i][j]$ ，所以 我们枚举 j 的通知 记录一个 满足  $b[k] < a[i]$  情况下的  $f[i - 1][k]$  的最大值进行转移即可。

【核心代码】

```
for(int i=1;i<=n;i++)
{
    int val=0; // val 表示当前最大子状态
    for(int j=1;j<=m;j++)
    {
        if(a[i]==b[j]) f[i][j]=val+1;
        else f[i][j]=f[i-1][j];
        // j 即将增大为 j+1，检查 j 能否更新 val
        if(b[j]<a[i]) val=max(val,f[i-1][j]);
        ans=max(ans,f[i][j]);
    }
}
```