

**题 1、IP 网络**

ip 的四个部分分别处理，每一部分找出最大值的最小值，然后判断是其二进制最后几位不同，得出子网掩码；

用任意一个 IP 与子网掩码进行按位与运算得出最小 IP；

**题 2、排列**

递推算法

设  $f(i, j)$  表示  $1..i$  的所有排列中，含有  $j$  个“<”的排列数。则可得递推方程：

$$f(i, j) = f(i-1, j) * (j+1) + f(i-1, j-1) * (i-j)$$

**题 3、猴子捞月**

算法 1、最优路径算法

这道题可以用最短路完成。猴子的手看成一条无向边，权值表示什么时候猴子放这条手，不放值为无穷大。一个猴子是否掉下的充要条件就是与 1 号猴子不连通，换句话说，1 到  $i$  号猴子的路径上最后断的一条使 1 与  $i$  不连通的边的权值就是  $i$  的掉落时间，所以  $i$  的掉落时间就是从 1 到  $i$  的路径上的最小边的最大值。可以用最短路算法解决（迪杰斯特拉算法  $O(n \log m)$ ），SPFA 可能会卡掉 1 个点！

算法 2、并查集

设带权并查集： $tt[i]$  表示并查集中点  $i$  与  $fa[i]$  连通的最早时刻。

然后倒着来处理放手信息，从而把放手转换为抓住（时光倒流）。也就是说，原图中一开始没有任何放手集合的边，然后从第  $m$  到  $m-1, \dots, 1$  一条边依次加入，每次加入后那些点原先和 1 不连通，但是现在连通了，则这个点“掉下来”的时间就是现在连通的时间！显然合并就用并查集（设为  $s$ ）检验撒。

时间复杂度为  $O(n+m \cdot A(m, n))$ ，其中  $A(m, n)$  为 Ackerman 函数的增长极为缓慢的逆函数！