Solution

idy002

July 30, 2017

1 treasure

1.1 10%

用 $C(n,m) = C(n-1,m-1) + C(n-1,m)$ 递推即可.
复杂度 $O(n^2)$

1.2 30%

预处理 0 到 n 的阶乘及其在模 p 意义下的逆元, 用 $C(n,m) = \frac{n!}{m!(n-m)!}$ 计算即可.
复杂度 $O(n)$.

1.3 60%

用 lucas 定理把原问题转换为上面的问题.
复杂度 $O(p + log(n))$.

1.4 100%

对于每个 p, 先用上面的方法算出 $C(n,m)$ 其模 p 的值, 就得到了 k 个形如:

$$x_i \equiv a_i \quad (mod\ p_i)$$

的同余方程, 用中国剩余定理合并一下即可. 中间可能会用到快速乘.
复杂度 $O(k(p + log(n)))$

2 kand

2.1 10%

暴力 dfs 所有 k 组合即可.
复杂度 $O(C(n,k))$.

2.2 30%

考虑动态规划, 状态 $dp[i][j][S]$ 表示前 i 个数选了 j 个出来, 其按位与为 S 的方案数, 每次 $O(1)$ 转移.
复杂度 $O(n^2 2^k)$, 其中 k 为最大位数.

2.3 60%

考虑容斥原理.
我们用 $cnt1[S]$ 表示值为 S 的数的个数, 通过它可以计算出 $cnt2[S]$, 表示值"包含"S 的数的个数 (a 包含 b 当且仅当 $a\&b = b$).
这一步可以先枚举一个 s, 然后枚举 $U - s$ 的子集 ss, 将所有 $cnt1[s|ss]$ 加到 $cnt2[s]$ 中即可, 复杂度 (3^k),k 为最大位数.
然后通过 $cnt2[S]$ 可以计算出 $cnt3[S]$, 表示从原来的 n 个数中选择 k 个并取交后, 该值包含 S 的方案数. 显然有 $cnt3[s] = comb(cnt2[s], k)$.
然后计算 $cnt4[S]$, 表示取 k 个数取交后结果为 S 的方案数. 和第二步类似, 只是每次从加变成减 (从大到小枚举).
以上第一步和第三步复杂度为 $O(3^k)$, 第二步为 $O(2^k)$
总的复杂度为 $O(3^k + n)$

2.4 100%

总的思想和上面一致, 只是我们可以将第一步和第三步复杂度优化到 $O(k2^k)$.
枚举 i 从 0 到 $k-1$, 枚举到 i 时, 保证 $[0, i-1]$ 已经加完,$[i, k-1]$ 位还没有动, 然后枚举 $U^(1 << i)$ 的子集 s, 将 $s|(1 << i)$ 加到 s 上去.
这步需要自己想想合理性 (建议结合代码).
总的复杂度 $O(k2^k + n)$.

3 solar