NOI模拟赛2 题解

saffah

2015 年 5 月 29 日

1　函数计算

1.1　暴力算法

首先要搞懂这道题在干什么：有一个以h_1, h_2两项开头的广义Fibonacci数列（模n意义下）和m次询问，每次询问一个区间的最大值，最后输出所有询问的答案之和。

预处理出数列并暴力处理询问，时间复杂度$O(r+mk)$。可以通过第1个测试点。

1.2　不太暴力的算法

后5个测试点中r都非常大，不能暴力预处理数列。

然而Fibonacci数列在模n意义下是有循环节的，并且循环节长度最坏不会超过$6n$，随机情况下更小。$O(n)$预处理出循环节长度和第一个循环节中的所有数，就可以$O(1)$查询数列的任何一个位置。可以通过第3个测试点。

暴力处理询问，时间复杂度$O(n+mk)$，时间上有压力。不妨预处理出所有询问的答案，时间复杂度$O(nk+m)$。可以通过第2个测试点。

用Sparse Table或线段树进行预处理，视常数大小期望得分40～100分。

1.3　标准算法

令a_i为原数列，b_i表示区间$[i, i+k]$的询问的答案。考虑这样处理：

首先暴力计算a_1，并找到最大值的位置i。然后从i向后找至多k个位置，找到第一个大于等于a_i的位置j，则b_2到b_{j-k-1}都应该是a_i，如此一直找下去。如果找了k个位置都没有找到更大的，那么暴力进行重算。

对于随机数据，复杂度显然是能保证的。期望得分100分。

2 机器人

2.1 暴力算法

期望得分25～30分。

2.2 不暴力的算法

考虑$n = d = 2^k \le 262144$的情况。

令$\omega = \cos\alpha + i\sin\alpha$。那么编号为$i$的最终位置$X_i = \sum_{j=0}^{n-1} a_j \omega^{ij}$。（我们把$a$的所有元素前移一位，下标改为从0到$n-1$）

令$f(x) = \sum_{i=0}^{n-1} a_i x^i$，那么$X_i = f(\omega^i)$。实际上就是在求一个多项式在$\omega^0$到$\omega^{n-1}$的值。在$n = d = 2^k$的情况下，可以用FFT解决，时间复杂度$O(n\log n)$。

对于$n \ne d = 2^k$的情况，可以在a序列的末尾补0或在超过d以后再次覆盖到序列的开头即可。时间复杂度$O(n + d\log d)$。可以拿到中间的40分。

对于$d \ne 2^k$的情况，可以套用经典的多项式在多点求值的方法，时间复杂度$O(n\log^2 n)$，期望得分70～100分。

2.3 标准算法

考虑$X_i = \sum_{j=0}^{n-1} a_j \omega^{ij}$，其中$ij = -\frac{(i-j)^2}{2} + \frac{i^2}{2} + \frac{j^2}{2}$，则：

$$X_i = \sum_{j=0}^{n-1} a_j \omega^{-\frac{(i-j)^2}{2}} \omega^{\frac{i^2}{2}} \omega^{\frac{j^2}{2}} = \omega^{\frac{i^2}{2}} \sum_{j=0}^{n-1} \left(a_j \omega^{\frac{j^2}{2}} \right) \omega^{-\frac{(i-j)^2}{2}}$$

令$A_i = a_i \omega^{\frac{i^2}{2}}, B_i = \omega^{-\frac{i^2}{2}}$，那么$X_i = \sum_{j=0}^{n-1} A_j B_{i-j}$，是一个卷积形式，可以用FFT计算。

时间复杂度$O(n\log n)$，期望得分100分。

3 游戏总分

3.1 暴力算法

超级大暴力可以得到10分。

考虑动态规划：设$f(t, S, T)$表示，当前小P手里的玩具集合为S，经过t轮后集合变为T的方案数。这个是很好dp出来的。对于每次询问，枚举最后的集

合，算出方案数与得分的乘积就可以了。时间复杂度$O(tn4^n+q2^n)$，期望得分25分。

3.2 不太暴力的算法

定义两个集合S,T的对称差$S \text{ xor } T=\{x|(x\in S) \text{ xor } (x\in T)\}$。容易发现$f(t,S,T)=f(t,\emptyset,S \text{ xor } T)$。

套用上面的算法，状态数从$t4^n$变为了$t2^n$，我们可以做到$O(tn2^n+q2^n)$，期望得分40分。

然而我们不难发现，其实只需要t和$|S \text{ xor } T|$都相等就能保证f相等。状态数变为了tn，总时间$O(tn+q2^n)$，期望得分50分。

这时发现我们的瓶颈是处理询问。不妨对于每个S和d，预处理出所有满足$|S \text{ xor } T|=d$的T的得分总和$g(S,d)$，这样对于每次询问我们不需要枚举所有集合，只需$O(n)$枚举d就可以了。

暴力枚举S,T来更新$g(S,d)$，总时间$O(4^n+tn+qn)$，期望得分75分。

3.3 标准算法

最后的问题就是快速处理出所有的$g(S,d)$。我们用A_i表示第i个玩具。考虑动态规划：$h(i,S,d)$表示与S的距离为d的集合的得分总和，但要求只有S的前i个元素能够发生变化。容易写出$h(i,S,d)=h(i-1,S,d)+h(i-1,S \text{ xor } \{A_d\},d-1)$，而$g(S,d)=h(n,S,d)$。用这种算法就可以$O(n^22^n)$求出所有的$g(S,d)$。如果在时间或空间上有压力，滚动数组优化即可。

期望得分100分。