《招新》解题报告

周雨扬

北京大学

2021 年 5 月 16 日

题目大意

有 n 个在 $0 \sim m$ 内等概率均匀随机变量 a_i。不妨假设将 a_i 从小到大排序过后的结果为 $x_1, x_2, \cdots, x_n$。

询问有多大的概率使得满足如下条件:

$$\forall i \geq 3, x_i \geq x_{i-2} + k$$

答案对质数 p 取模。

$1 \leq n \leq 50, 1 \leq k \leq m \leq 150, 10^8 \leq p \leq 10^9$。

出题背景

算协招新，利用学 OI 学到的博弈技巧，~~把工作人员正反手打爆~~。

出题背景

算协招新，利用学 OI 学到的博弈技巧，~~把工作人员正反手打爆~~。
然后客串工作人员~~把其余学生正反手打爆~~。

出题背景

算协招新，利用学 OI 学到的博弈技巧，~~把工作人员正反手打爆~~。
然后客串工作人员~~把其余学生正反手打爆~~。
欢迎在进入北大后加入算协出题！

大前提

不难发现将随机方式改成如下，不会改变答案的期望：

- 在所有满足 $0 \leq x_1 \leq x_2 \leq \cdots \leq x_n \leq m$ 的 n 元组中等概率选取一个，满足题目所描述的条件的概率。

算法 1

我会推导二元概率函数！

算法 1

我会推导二元概率函数！

假设 $f_{i,j,k}(x,y)$ 表示考虑了前 i 个随机变量 x_i，x_i 整数部分为 j, x_{i-1} 整数部分为 k 的二元概率函数。

$f_{i,j,k}(x,y)$ 满足 x_i 整数部分为 j, 小数部分落在 $[l_1,r_1]$, x_{i-1} 整数部分为 k, 小数部分落在 $[l_2,r_2]$ 的概率恰好为 $\int_{x=l_1}^{r_1}\int_{y=l_2}^{r_2} f_{i,j,k}(x,y)\mathrm{d}x\mathrm{d}y$

算法 1

我会推导二元概率函数！

假设 $f_{i,j,k}(x,y)$ 表示考虑了前 i 个随机变量 x_i， x_i 整数部分为 j, x_{i-1} 整数部分为 k 的二元概率函数。

$f_{i,j,k}(x,y)$ 满足 x_i 整数部分为 j, 小数部分落在 $[l_1,r_1]$, x_{i-1} 整数部分为 k, 小数部分落在 $[l_2,r_2]$ 的概率恰好为 $\int_{x=l_1}^{r_1}\int_{y=l_2}^{r_2} f_{i,j,k}(x,y)\mathrm{d}x\mathrm{d}y$

转移可以通过二元函数的积分的偏导和积分进行状态的转移。

通过转移函数我们可以归纳证明，所有 $f_{i,j,k}(x,y)$ 均为次数不超过 $2n$ 的关于 x,y 的二元多项式函数。因此直接维护各项系数即可。

算法 1

我会推导二元概率函数！

假设 $f_{i,j,k}(x,y)$ 表示考虑了前 i 个随机变量 x_i，x_i 整数部分为 j，x_{i-1} 整数部分为 k 的二元概率函数。

$f_{i,j,k}(x,y)$ 满足 x_i 整数部分为 j, 小数部分落在 $[l_1,r_1]$, x_{i-1} 整数部分为 k, 小数部分落在 $[l_2,r_2]$ 的概率恰好为 $\int_{x=l_1}^{r_1}\int_{y=l_2}^{r_2} f_{i,j,k}(x,y)\mathrm{d}x\mathrm{d}y$

转移可以通过二元函数的积分的偏导和积分进行状态的转移。

通过转移函数我们可以归纳证明，所有 $f_{i,j,k}(x,y)$ 均为次数不超过 $2n$ 的关于 x,y 的二元多项式函数。因此直接维护各项系数即可。

时间复杂度 $O(m^2n^3)$ 或 $O(m^2n^4)$，如果采用前缀和优化可以实现到 $O(m^2n^3)$。代码实现难度较高。

算法 2

我会简单 DP!

算法 2

我会简单 DP!

如果我们已经知道了 $x_1, \cdots, x_n$ 的小数部分的相对顺序，则整数部分的方案计数是一个非常简单的 DP。

具体的设 $f_{i,j,k}$ 表示确定了前 i 个数字的整数部分，x_i, x_{i-1} 整数部分分别是 j, k 的概率。转移非常简单。

算法 2

我会简单 DP!

如果我们已经知道了 $x_1, \cdots, x_n$ 的小数部分的相对顺序，则整数部分的方案计数是一个非常简单的 DP。

具体的设 $f_{i,j,k}$ 表示确定了前 i 个数字的整数部分，x_i, x_{i-1} 整数部分分别是 j, k 的概率。转移非常简单。

最后除以满足 $0 \le x_1 \le x_2 \le \cdots \le x_n \le m$ 的方案数即可。

时间复杂度 $O(n! \times nm^2)$.

算法 3

我会改进算法 2!

算法 3

我会改进算法 2!

这里我们发现算法 2 的转移仅仅和 x_i, x_{i-1}, x_{i+1} 的小数部分的相对顺序有关。这里我们只需要记录 x_i, x_{i-1} 的小数部分在前 i 个数字小数部分的排名即可。

具体的设 $f_{i,j,k,l,m}$ 表示确定了前 i 个数字的整数部分，x_i, x_{i-1} 整数部分分别是 j, k , 小数部分的相对排名为 l, m 的概率。转移类似于算法 2，也非常简单。

算法 3

我会改进算法 2!

这里我们发现算法 2 的转移仅仅和 x_i, x_{i-1}, x_{i+1} 的小数部分的相对顺序有关。这里我们只需要记录 x_i, x_{i-1} 的小数部分在前 i 个数字小数部分的排名即可。

具体的设 $f_{i,j,k,l,m}$ 表示确定了前 i 个数字的整数部分，x_i, x_{i-1} 整数部分分别是 j, k , 小数部分的相对排名为 l, m 的概率。转移类似于算法 2，也非常简单。

时间复杂度 $O(n^4m^3)$。如果使用前缀和优化可以优化至 $O(n^3m^2)$.

算法 4

我会改进算法 3!

算法 4

我会改进算法 3!

当 $n \le 2$ 的时候答案为 1. 这种情况可以扔了。

当 $\lceil (\frac{n+1}{2} - 1) \rceil k \ge m$ 的时候不论怎么排列都不合法，答案为 0.

如果把这两种边界情况挖掉，状态个数能否得到优化？

算法 4

乍一看状态并没有得到优化，但是我们观察算法 3 的转移，不难发现当 x_{i-1} 整数部分和 x_i 整数部分之差大于 k 的时候，不同状态的转移完全一模一样。

算法 4

乍一看状态并没有得到优化，但是我们观察算法 3 的转移，不难发现当 x_{i-1} 整数部分和 x_i 整数部分之差大于 k 的时候，不同状态的转移完全一模一样。

这提示我们可以只记录 x_i，x_{i-1} 整数部分之差对 $k+1$ 取较小值的值，而不用存储 $1 \sim m$ 的值。

将状态优化为 $f_{i,j,k,l,m}$ 表示确定了前 i 个数字的整数部分，x_i 整数部分分别是 j，x_i，x_{i-1} 整数部分之差为 k，小数部分部分的相对排名为 l, m 的概率。转移仍然和算法 3 类似。

算法 4

此时状态个数为 $O(n^3km)$ 。但是由于当且仅当 $nk \leq 2m$ 的时候有解，因而实际状态数为 $O(n^2m^2)$。

利用前缀和优化，总时间复杂度可以优化为 $O(n^2m^2)$，足以通过。

看上去很复杂，但实际上不知道比算法 1 好写到哪里去了。