# 有趣的游戏解题报告

浙江省绍兴市第一中学洪华敦

## 1 试题来源

https://www.51nod.com/onlineJudge/questionCode.html#!problemId=1599

## 2 试题大意

有$n$个人在玩游戏，游戏的规则是这样的:

(1)随机选择一个未出局的人$x$

(2)$x$对剩下的未出局的人都做一次攻击，一个人受到攻击后，有$(1-p)$的概率受伤且出局

(3)$x$出局

重复以上步骤直到所有人出局

现在$ZYB$想知道，一个人不受伤出局，且一共受到$k$次攻击的概率是多少

为了避免精度误差，你需要输出答案在模$1e9+7$域下的值

一共有$Q$次询问，每次询问的$n$一样，$1 \le n \le 10^6$，$1 \le \sum k \le 10^6$

保证$p^0, p^1...p^n$互不相等

## 3 算法介绍

### 3.1 暴力动态规划

考虑动态规划的做法，暴力做法是$f[i][j]$记剩下$i$个人，有$j$次攻击的概率，转移时需要枚举多少人受伤出局，时间复杂度是$O(n^3)$

### 3.2 较优的动态规划

本题有一个十分重要的性质：每个人本质上是相同的

之前的动态规划之所以是$n^3$的，是因为这个序列的数量变化不稳定，如果每次变化稳定，转移就可以变成$O(1)$的了

我们假定第二步的受伤不出局，然后在第一步选人时，如果选的受伤的就直接执行第三步，这样显然概率还是等价的

就像一堆0，一堆1，你选到第一个0是你想要的的概率显然还是$\frac{1}{|S_0|}$

那么就可以设计出新的动态规划方程$f[i][j]$表示还剩下$i$个人，前面一共$j$次攻击

新加入一个人时，考虑他在$j$次攻击下存活的概率即可

即$f[i][j] = f[i-1][j] * (1-p^j) + f[i-1][j-1] * p^j$

最后答案就是$\frac{\sum_{i=0}^{n-1} f[i][k] * p^k}{n}$

这样时间复杂度是$O(n^2)$的

### 3.3 多项式

考虑用多项式优化这个$Dp$

令

$$g[i][j] = f[i][j] * p^j$$

则有

$$g[i][j] = g[i-1][j-1] * p^j + g[i-1][j] * (1-p^j)$$

令多项式$h_i(x)$为对于$0 \le j \le n$满足$[x^j]h_i(x) = g[j][i]$的多项式

则有

$$h_k(x) = h_{k-1}(x) * x * p^k + h_k(x) * x * (1-p^k)$$

移项得

$$h_k(x) = \frac{h_{k-1}(x) * x * p^k}{1-(1-p^k) * x}$$

显然有

$$h_0(x) = 1$$

所以

$$h_k(x) = \prod_{i=1}^{k} \frac{x * p^i}{1-(1-p^i) * x}$$

$$h_kx = x^k * p^{\frac{k*(k+1)}{2}} * \prod_{i=1}^{k} \frac{1}{1-(1-p^i)*x}$$

设

$$\prod_{i=1}^{k} \frac{1}{1-(1-p^i)*x} = \sum_{i=1}^{k} \frac{A_i}{1-(1-p^i)*x}$$

现在

$$ans_k = \sum_{i=0}^{n-1} [x^i]h_k(x)$$

$$ans_k = \sum_{j=0}^{n-1} [x^j]x^k * p^{\frac{k*(k+1)}{2}} * \sum_{i=1}^{k} \frac{A_i}{1-(1-p^i)*x}$$

$$ans_k = \sum_{j=0}^{n-1} [x^j](x^k * p^{\frac{k*(k+1)}{2}} * \sum_{i=1}^{k} A_i * \sum_{v=0}^{oo} (1-p^i)^v * x^v)$$

$$ans_k = p^{\frac{k*(k+1)}{2}} \sum_{j=0}^{n-1-k} \sum_{i=1}^{k} A_i * (1-p^i)^j = p^{\frac{k*(k+1)}{2}} \sum_{i=1}^{k} A_i * \frac{(1-p^i)^{n-k}-1}{(1-p^i)-1}$$

于是只要求出Ai我们就可以在O(klogk)的时间内算出答案，现在我们来计算$A_i$

根据定义，有：

$$\frac{A_i}{1-(1-p^i)x} + \sum_{i \neq j} \frac{A_j}{1-(1-p^j)x} = \prod_{i=1}^{k} \frac{1}{1-(1-p^i)x}$$

$$A_i + (1-(1-p^i)x) * \sum_{i \neq j} \frac{A_j}{1-(1-p^j)x} = \prod_{i \neq j} \frac{1}{1-(1-p^j)x}$$

令

$$x = \frac{1}{1-p^i}$$

则有

$$A_i = \prod_{i \neq j} \frac{1}{1-\frac{1-p^j}{1-p^i}} = \prod_{i \neq j} \frac{1-p^i}{p^j-p^i} = (\frac{1-p^i}{p^i})^{k-1} * \prod_{i \neq j} \frac{1}{p^{j-i}-1}$$

$O(n)$预处理$\prod_{i=1}^{n} \frac{1}{p^i-1}$就可以在$O(klogk)$的时间内求出所有$A_i$了

于是这题就解决了

时间复杂度$O(klogk)$