Solution 8

1、假定有n个不同的数以串行的方式逐个试图写入某个固定的内存单元。要求，在写入之前，每个元素会将自身的值和目前内存单元的值比较一下，如果大于目前内存单元的值，则写入；否则将自身的值直接抛弃（不写入）。假定该内存单元的初始值为全局最小值，在本题里可以假设为0，求实际发生写入的期望值。

更准确地来说，给定一个长度为n的序列$a_1, a_2, \cdots, a_n$，记f_i为a的第i个前缀的最大值，即$f_i = \max\{a_1, a_2, \cdots, a_i\}$，求$f$中不同值数量的期望，即求$E[|\{x \in f\}|]$。

(a) 假设a是一个随机的1到n的排列，求f中不同值数量的期望。

例：$a = [1,3,2], f = [1,3,3]$，则f中不同值的数量为2

考虑最后一位，对答案有贡献当且仅当$a_n = n$，$\Pr[a_n = n] = \frac{1}{n}$，前$n-1$位对答案的贡献和最后一位无关，可以看做一个$n-1$阶的问题。

因此答案$= \sum_{i=1}^{n} \frac{1}{i} = \Theta(\log n)$

(b) 假设a中每个数互相独立且服从$[1,n]$的离散均匀分布（序列中可能出现重复的数），求f中不同值数量的期望，你不需要计算出具体的期望，而是需要给出期望的确界，即求出$E = \Theta(g(n))$。

例：$a = [1,1,2], f = [1,1,2]$，则f中不同值的数量为2例：$a = [1,1,2], f = [1,1,2]$，则f中不同值的数量为2

同样考虑最后一位，对答案有贡献当且仅当最后一位>之前所有的数。
$\Pr[a_n > f_{n-1}] = \sum_{i=1}^{n} \frac{1}{n} (\frac{i-1}{n})^{n-1}$。

答案$= \frac{1}{n} + \sum_{j=1}^{n} \sum_{i=1}^{n} \frac{1}{n} (\frac{i-1}{n})^{j-1}$($+\frac{1}{n}$是因为第1位少算了$\frac{1}{n}$)

$$
\begin{aligned}
&\overset{\text{交换求和顺序}}{=} \frac{1}{n} + \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{n} (\frac{i-1}{n})^{j-1} \\
&= \frac{1}{n} + \sum_{i=1}^{n} \frac{1 - (\frac{i-1}{n})^n}{1 - \frac{i-1}{n}} \\
&= \frac{1}{n} + \sum_{i=1}^{n} \frac{1 - (1 - \frac{n-i+1}{n})^n}{n-i+1} \\
&\overset{t=n-i+1}{=} \frac{1}{n} + \sum_{t=1}^{n} \frac{1 - (1 - \frac{t}{n})^n}{t} \\
&\underset{n\to\infty}{\approx} \sum_{t=1}^{n} \frac{1 - e^{-t}}{t} \\
&\geq (1 - e^{-1}) \sum_{i=1}^{n} \frac{1}{i} = \Omega(\log n) \\
&\leq 1 \sum_{i=1}^{n} \frac{1}{i} = O(\log n)
\end{aligned}
$$

因此答案=$\Theta(\log n)$

2、假设有n种不同的颜色，每种颜色分别有1个桶和1个球。现在让一个人蒙上眼睛后随机的将球放入桶里，并假定每个桶里只能放1个球。问最后桶和球的颜色匹配的数量的期望值是多少？

更准确地来说，给定一个长度为n的随机排列p，求$E[\sum_{i=1}^{n} [p[i] = i]]$。

每个球都有$\frac{1}{n}$的概率放到正确的位置，因此$E[\sum_{i=1}^{n}[p[i]=i]]=n\cdot\frac{1}{n}=1$

3、**Oblivious Routing on the Hypercube**

算法：1、每个点i将数据包传输到一个随机点R_i

2、$5d$轮后，将数据包从当前所在的点传输到π_i

Theorem 1：随机算法在$10d$轮内传输完成的概率$\Pr\geq 1-\frac{2}{n}$

Claim 2：假如$i\to R_i$和$j\to R_j$的路径存在公共路径，那么这样的公共路径最多只有一段。

(a) 请给出Claim 2的证明：

$i\to R_i$和$j\to R_j$一旦分开，意味着R_i和R_j的这一个二进制位不同，那么之后一定不会再相交。

Claim 3：记$i\to R_i$的路径为P_i，和P_i存在公共路径的P_j的集合为S_i。则i到达R_i的时间最迟为$|P_i|+|S_i|$。

- 如果$S_i=\emptyset$，显然只要$|P_i|$的时间
- 我们希望将每次等待的时间分摊到最多一个$P_j\in S_i$，那么$i\to R_i$最多等待$|S_i|$的时间
- 根据以下方式定义lag：对于P_i中的第k条边e_k，在t时刻，位于e_k起点w_{e_k}的包j拥有lag$=t-k$。需要注意的是lag的定义都是基于P_i的，而不是基于P_j的。
- 当t时刻，包i在第k个点被包j阻塞时，向包j发送一个$(t-k)$的token，那么包i发送的每个token的值都是唯一的（每次被阻塞，token的值+1）
- 当t时刻，包j在第k个点被包x阻塞时，如果包j有token，则将这个token传给包x
- (b) 证明：包x在此之前一定没有token，且同一时刻t，包x最多从一个包j处接受token
- 因为包k接受的token=包k的lag，而每个token都是唯一的，所以包k不会有超过1个token。
- (c) 证明：每次包i交给包j一个新的token时，包j一定没有token。
- 因为包i发的每个token的lag都是唯一的，且等于当前包j的lag，假设包j之前有一个token，那这个token的lag一定更小，意味着包j一定被某个包k阻塞过，此时包j会把token传给包k，因此包j上一定没有token。
- （Hint: 从lag的角度考虑）
- 因此每个包做多接受1个token，一共最多阻塞$|S_i|$次
- 总时间=移动的时间+阻塞的时间$\leq|P_i|+|S_i|$

下面证明$|S_i|$的上界：

Claim 4: $\Pr[|S_i|\geq 4d]\leq e^{-2d}$

(d) 固定任意一条路径P_i，证明$E[|S_i|]\leq\frac{d}{2}$

每条边平均被$\frac{\text{点数}\cdot\text{平均路径长度}}{\text{边数}}=\frac{n\cdot\frac{d}{2}}{n\cdot d}=\frac{1}{2}$条路径覆盖。$E[|S_i|]\leq d\cdot\frac{1}{2}=\frac{d}{2}$

(e) 使用霍夫丁不等式证明Claim 4

$\Pr[|S_i|\geq 4d]\leq\exp\{-\frac{(4d-\mu)^2}{2\mu+(4d-\mu)}\}\leq\exp\{-2d\}$

(f) 证明：在$5d$轮内所有$i\to R_i$传输完成的概率$\Pr\geq 1-\frac{1}{n}$(Hint: use union bound)

$\Pr\geq 1-2^d\cdot e^{-2d}\geq 1-e^{-d}\geq 1-\frac{1}{n}$

算法的后半部分证明类似。