数据结构与算法第七次作业

麻超　201300066

南京大学人工智能学院

2021 年 11 月 11 日

目录

1 Problem 1

1.1 a

证明:Q_k 表示正好有 k 个关键字被散列到某一特定槽中的概率, 则有 $\binom{n}{k}$ 种方法选出 n 次试验中哪 k 次成功. 被散列到特定槽 X0 即视为成功, 其成功率就是 1/n, 相反则视为失败. 概率为 $1-\frac{1}{n}$. 其概率服从伯努利二项分布, 即概率为 $(\frac{1}{n})^k$ 在这个过程之后, 剩下了 n-k 个关键字需要散列到 n-1 个槽中, 概率为 $(\frac{n-1}{n})^{n-k}$, 所以总的概率

$$
\begin{aligned}
Q_k =&(\frac{1}{n})^k(\frac{n-1}{n})^{n-k}\binom{n}{k}\\
=&\frac{1}{n^k}\frac{(n-1)^{n-k}}{n^{n-k}}\frac{n!}{k!(n-k!)}\\
=&(1-1/n)^{n-k}\cdot\frac{n(n-1)...(n-k+1)}{n^k}\cdot\frac{1}{k!}\\
\leq&\frac{n(n-1)...(n-k+1)}{n^k}\cdot\frac{1}{k!}\\
\leq&\frac{n\cdot n\cdot n\cdots n(total-k-times)}{n^k}\frac{1}{k!}\\
\leq&\frac{1}{k!}\\
<&\frac{e^k}{k^k}
\end{aligned}
$$

1.2 b

被散列到槽中最多的关键字数量视为一个事件 $X=X_1\cup X_2\cup...\cup X_n$，其中 X_i 为某个关键字被散列到槽 i 的事件。所以根据布尔不等式有

$$P_k\leq P(x_1)+P(X_2)+...+P(x_n)=nQ_k$$

1.3 c

如果需要证明 $P_k<\frac{1}{n^2}$, 由第二问的结论, 需要证明 $Q_k<\frac{1}{n^3}$, 即需要证明存在常数 c>1, 使得 $Q_{k_0}<\frac{1}{n^3}$ 对 $k_0=c\lg n/\lg\lg n$ 成立.

由于 $Q_k<\frac{e^k}{k^k}$

所以只需要证明 $n^3\leq k!$, 即 $3<\frac{k_0(\lg k_0-\lg e)}{\lg n}$

即

$$
\begin{aligned}
3 <& \frac{c\lg n}{\lg n \lg\lg n}(\lg \frac{c\lg n}{\lg\lg n} - \lg e) \\
=& \frac{c}{\lg\lg n}(\lg c + \lg\lg n - \lg\lg\lg n - \lg e) \\
=& c(1 + \frac{\lg c - \lg e}{\lg\lg n} - \frac{\lg\lg n}{\lg\lg\lg n})
\end{aligned}
$$

选择一个必要的 c 的值取决于 n 的大小, 对括号中表达式有 $\lim_{n\to\infty} A = 1$

存在一个 n_0, 当 $n \geq n_0$ 时,$A \geq 1/2$, 因此选择 $c > 6$, 此时必然有当 $n \geq n_0$ 时,$3 < cA$. 接下来需要说明 $n < n_0$ 的情况. 同时,n 一定是大于或等于 3 的, 因为如果不是的话,$\lg\lg 2 = \lg 1 = 0$, 不能作为除数.

当 $3 \leq n < n_0$ 时, 选择任何大到足以满足所有情况的 c, 即 $max_{3\leq n<n_0}\{c : 3 < cx\}$, 选择 c 和 6 之间相对较大的一个, 且必然存在这样的一个数字.(这里的原因还未研究清楚).

1.4 d

$$
\begin{aligned}
E(M) &= \sum_{i=1}^{\frac{c\lg n}{\lg\lg n}} i \times Pr(M = i) + \sum_{i=\frac{c\lg n}{\lg\lg n}+1}^{n} i \times Pr(M = i) \\
&< \sum_{i=1}^{\frac{c\lg n}{\lg\lg n}} \frac{c\lg n}{\lg\lg n} \times Pr(M = i) + \sum_{i=\frac{c\lg n}{\lg\lg n}+1}^{n} n \times Pr(M = i) \\
&= \frac{c\lg n}{\lg\lg n} \times Pr(M \leq \frac{c\lg n}{\lg\lg n}) + n \times Pr(M > \frac{c\lg n}{\lg\lg n})
\end{aligned}
$$

得证.

对于后一部分, 有

$$
Pr(M > \frac{c\lg n}{\lg\lg n}) = \sum_{k=\frac{c\lg n}{\lg\lg n}+1}^{n} Pr(M = k) \leq \sum_{k=\frac{c\lg n}{\lg\lg n}+1}^{n} \frac{1}{n^2} \leq n\frac{1}{n^2} = \frac{1}{n}
$$

所以

$$
\begin{aligned}
E(M) &= \frac{c\lg n}{\lg\lg n} \times Pr(M \le \frac{c\lg n}{\lg\lg n}) + n \times Pr(M > \frac{c\lg n}{\lg\lg n}) \\
&\le \frac{c\lg n}{\lg\lg n} + 1 \\
&= O(\frac{\lg n}{\lg\lg n})
\end{aligned}
$$

2 Problem 2

2.1 a

证明: 设字符串 x 的各个字符为 $x = c_{l-1}c_{l-2}...c_2c_1c_0$, 其中 c_i 为字符串 x 从右往左数的第 i 个字符, 用 n_i 表示 (是一个数字),l 为字符串的长度. 可以设 $\alpha = 2^p, \beta = 2^p - 1$

于是有

$$
\begin{aligned}
h(k) &= (\sum_{i=0}^{l-1}(n_i \cdot 2^{ip})) \mod (2^p - 1) \\
&= (\sum_{i=0}^{l-1}(n_i \cdot \alpha^i)) \mod \beta \\
&= (n_0 + n_1 \cdot \alpha + n_2 \cdot \alpha^2 + ... + n_{l-1} \cdot \alpha^{l-1}) \mod \beta \\
&= (n_0 \mod \beta + n_1 \cdot \alpha \mod \beta + n_2 \cdot \alpha^2 \mod \beta + ... + n_{l-1} \cdot \alpha^{l-1} \mod \beta) \mod \beta \\
&= (n_0 \mod \beta + (n_1 \mod \beta) \cdot (\alpha \mod \beta) \mod \beta + (n_2 \mod \beta) \cdot (\alpha^2 \mod \beta) \mod \beta \\
&+ ... + (n_{l-1} \mod \beta) \cdot (\alpha^{l-1} \mod \beta) \mod \beta) \mod \beta \\
&= ((n_0 \mod \beta) + (n_1 \mod \beta) + (n_2 \mod \beta) + ... + n_{l-1} \mod \beta) \mod \beta \\
&= (n_0 + n_1 + n_2 + ... + n_{l-1}) \mod \beta
\end{aligned}
$$

对字符串 x 和 y 来说, 由于只改变了每个字符的顺序, 所以其各个字符位之和不变, 故 $h(k)$ 不变, 即结果与所包含的字符顺序无关. 得证.

2.2 b

证明: 设经过 s 次查找后，回到了槽 $h_1(k)$. 于是就有 $(h_1(k) + sh_2(k)) \mod m = h_1(k) \mod m$. 此时因为 $sh_2(k)$ 是 m 的倍数, 所以 $m|sh_2(k)$ 由于 $gcd(m, h2(k)) = d$ 所以 $(m/d)|s(h_2(k)/d)$ 由于 $gcd(m/d, h_2(k)/d) = 1$ 所以 $m/d|s$ 故 $s \ge m/d$. 所以在回到

槽 $h_1(k)$ 之前, 必检查散列表总元素数的至少 $\frac{1}{d}$ 个. 当 d=1 时,$\frac{1}{d}=1$, 所以可能要检查整个散列表.

3 Problem 3

3.1 a

对于一个哈希表而言, 每一个位置的冲突数由该位置存储元素个数决定, 对于任意一个位置 p, 用 n_p 表示实际存储的元素数量, 用 X_p 表示冲突数, 则 $X_p = n_p - 1(n_p \geq 2)$. 定义 Y 为总冲突数 (也即题目所求的期望),$Y = X_1 + X_2 + ... + X_m$

易知 $X_i \in [0, n-1]$. 对于每一个元素, 被放到第 i 个位置的概率 $p_1 = \frac{1}{m}$, 放到其他位置的概率 $p_2 = 1 - \frac{1}{m}$. 每一个元素放到第 i 个位置的概率服从二项分布, 假设 b_i 表示该位置放了 i 个元素, 故 $b_i = \binom{m}{i} p_1^i p_2^{n-i}$

所以

$$E(X_i) = 0(b_0 + b_1) + 1b_2 + 2b_3 + ... + (n-1)b_n$$

而 n_p 的期望

$$E(n_p) = 0b_0 + 1b_1 + 2b_2 + ... + nb_n$$

两式相减可以得到该位置冲突的数目

$$E(n_p) - E(X_i) = b_1 + b_2 + ... + b_n = 1 - b_0 = 1 - p_2^n$$

所以进一步可以得到

$$E(X_i) = \frac{n}{m} + (\frac{m-1}{m})^n - 1$$

进而可以得到 Y 的期望, 即为所求期望

$$E(Y) = m \cdot E(X_i) = n - m + m \cdot (\frac{m-1}{m})^n$$

3.2 b

对于一个完全随机的哈希函数, 一共有 n 个关键字,m 个槽的情况下, 一个哈希函数是完美的即它是单射. 所以其概率

$$\begin{aligned}P(n) =& \frac{m}{m} \times \frac{m-1}{m} \times ... \times \frac{m+2-n}{m} \times \frac{m+1-n}{m} \\ =& \frac{m!}{m^n(m-n)!}\end{aligned}$$

3.3 c

由 c 可知, 对于一个随机的哈希函数, 其完美的概率 $P = \frac{m!}{m^n(m-n)!}$, 而多次选取哈希函数以求得一个完美的哈希函数, 符合二项分布的特征, 故 $E(X) = 1/p = \frac{m^n(m-n)!}{m!}$

3.4 d

由 b 和 c 可知, 其服从二项分布, 每一次尝试得到完美的哈希函数概率都不变, 于是尝试的前 N 次都不完美的概率是

$$P(N) = (1-p)^N = (\frac{m!}{m^n(m-n)!})^N$$

3.5 e

假设任取了 a 个哈希函数, 其中至少有一个哈希函数是完美的的概率是 $1 - \frac{1}{n}$, 也就是让不存在完美的哈希函数的概率是小于 $\frac{1}{n}$. 对于任意一个哈希函数而言, 其不是完美的的概率是 $p = 1 - \frac{m!}{m^n(m-n)!}$.

那么也就是说任取 a 个哈希函数, 其没有一个完美的概率是 p^a. 那么此时要求 $p^a < \frac{1}{n}$. 所以 $a < \log_p \frac{1}{n}$

4 Problem 4

当 i 严格为 2 的幂次时, 此次调用的代价是 i, 否则为 1, 那么对于有 n 个操作组成的操作序列而言, 共有最多 $\lg n$ 个操作的代价为 n, 其他的均为 1, 于是

$$\begin{aligned}
cost(n) = \sum_{i=1}^{n} c_i &\le n + \sum_{j=0}^{\lceil \lg n \rceil} \frac{n}{2^j} \\
&\le n + n + \frac{n}{2} + \frac{n}{4} + ... + 2 + 1 \\
&\le n + \frac{n - \frac{1}{2}}{1 - \frac{1}{2}} \\
&= n + (2n - 1) \\
&\le 3n
\end{aligned}$$

所以每个操作的摊还代价 $\overline{c_i} \le \frac{3n}{n} = 3$, 即摊还代价为 $O(1)$.

5 Problem 5

对于一个散列表, 如果插入和删除时不涉及散列表大小的调整, 那么其所需时间为 O(1). 假设它此时需要承担插入 n 个元素的任务, 由于散列表的大小 i 只有扩大 2 倍和缩小一半的操作, 所以其必定为 2 的幂次. 所以需要执行的所有代价

$$\begin{aligned}
cost(n) = &\sum_{i=1}^{n} c_i \\
=& n + \sum_{\frac{3}{4}2^i \le n} (\frac{3}{4}2^i - 1) \\
\le& n + \sum_{\frac{3}{4}2^i \le n} (\frac{3}{4}2^i) \\
\le& \frac{5}{2}n
\end{aligned}$$

所以其每个操作的摊还代价 $\overline{c_i} = \frac{cost(n)}{n} \le \frac{2.5n}{n} = 2.5$, 即其摊还代价为 O(1). 由于删除操作可以视为插入操作的逆操作, 故可得知对于任何插入和删除序列，每个操作的摊销时间仍然是 O(1).