习题2-12

TC 11.2-3, P11.2

CS 5.5-8,11

11.2-3

Professor Marley hypothesizes that he can obtain substantial performance gains by modifying the chaining scheme to keep each list in sorted order. How does the professor's modification affect the running time for successful searches, unsuccessful searches, insertions, and deletions?

如果有序,就可以使用二分查找代替线性查找来加速

成功查找,不成功查找:$\mathbf{O}(1+\log\frac{n}{m})$

插入:$\mathbf{O}(1+\log\frac{n}{m})$(需要先查找,找到待插入的位置)

删除:$\mathbf{O}(1)$

?

Successful search: $\Theta\left(1+\frac{\alpha}{2}\right)$

Unsuccessful search: $\Theta(1+\alpha)$

Insertion: $\Theta\left(1+\frac{\alpha}{2}\right)$

Deletion: $\Theta(1)$

11-2 Slot-size bound for chaining

Suppose that we have a hash table with n slots, with collisions resolved by chaining, and suppose that n keys are inserted into the table. Each key is equally likely to be hashed to each slot. Let M be the maximum number of keys in any slot after all the keys have been inserted. Your mission is to prove an $O(\lg n / \lg\lg n)$ upper bound on $\mathrm{E}[M]$, the expected value of M.

a. Argue that the probability Q_k that exactly k keys hash to a particular slot is given by

$$Q_k = \left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k} \binom{n}{k} .$$

首先选出k个关键字,共有$\binom{n}{k}$种选法,对于每一种情况,$(\frac{1}{n})^k$为被散列到特定槽中的概率,$(\frac{1}{n})^{n-k}$为没有散列到该槽中的概率,因此

$$Q_k = (\frac{1}{n})^k(1 - \frac{1}{n})^{n-k}\binom{n}{k}$$

b. Let P_k be the probability that $M = k$, that is, the probability that the slot containing the most keys contains k keys. Show that $P_k \le nQ_k$.

c. Use Stirling's approximation, equation (3.18), to show that $Q_k < e^k / k^k$.

d. Show that there exists a constant $c > 1$ such that $Q_{k_0} < 1/n^3$ for $k_0 = c \lg n / \lg\lg n$. Conclude that $P_k < 1/n^2$ for $k \ge k_0 = c \lg n / \lg\lg n$.

e. Argue that

$$\mathrm{E}[M] \le \Pr\left\{M > \frac{c \lg n}{\lg\lg n}\right\} \cdot n + \Pr\left\{M \le \frac{c \lg n}{\lg\lg n}\right\} \cdot \frac{c \lg n}{\lg\lg n} .$$

Conclude that $\mathrm{E}[M] = O(\lg n / \lg\lg n)$.

11-2 Slot-size bound for chaining

Suppose that we have a hash table with n slots, with collisions resolved by chaining, and suppose that n keys are inserted into the table. Each key is equally likely to be hashed to each slot. Let M be the maximum number of keys in any slot after all the keys have been inserted. Your mission is to prove an $O(\lg n / \lg\lg n)$ upper bound on $\mathrm{E}[M]$, the expected value of M.

a. Argue that the probability Q_k that exactly k keys hash to a particular slot is given by

$$Q_k = \left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k} \binom{n}{k} .$$

首先选出k个关键字,共有$\binom{n}{k}$种选法,对于每一种情况,$(\frac{1}{n})^k$为被散列到特定槽中的概率,$(\frac{1}{n})^{n-k}$为没有散列到该槽中的概率,因此

$$Q_k = (\frac{1}{n})^k(1 - \frac{1}{n})^{n-k}\binom{n}{k}$$

b. Let P_k be the probability that $M = k$, that is, the probability that the slot containing the most keys contains k keys. Show that $P_k \le nQ_k$.

定义事件

- A_i: 第i个slot拥有最多的keys
- B_i: 第i个slot恰好具有k个keys
- $C_i: A_i \wedge B_i$
- $D: M = k$

$$P(D) = P_k = P(C_1 \cup C_2 \ldots \cup C_n)$$

$$\le P(C_1) + P(C_2) + \cdots P(C_n)$$

$$\le P(B_1) + P(B_2) + \cdots P(B_n)$$

$$\le nQ_k$$

11-2 Slot-size bound for chaining

Suppose that we have a hash table with n slots, with collisions resolved by chaining, and suppose that n keys are inserted into the table. Each key is equally likely to be hashed to each slot. Let M be the maximum number of keys in any slot after all the keys have been inserted. Your mission is to prove an $O(\lg n / \lg\lg n)$ upper bound on $\mathrm{E}[M]$, the expected value of M.

a. Argue that the probability Q_k that exactly k keys hash to a particular slot is given by

$$Q_k = \left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k} \binom{n}{k}.$$

首先选出k个关键字,共有$\binom{n}{k}$种选法,对于每一种情况,$(\frac{1}{n})^k$为被散列到特定槽中的概率,$(\frac{1}{n})^{n-k}$为没有散列到该槽中的概率,因此

$$Q_k = (\frac{1}{n})^k(1-\frac{1}{n})^{n-k}\binom{n}{k}$$

b. Let P_k be the probability that $M = k$, that is, the probability that the slot containing the most keys contains k keys. Show that $P_k \le nQ_k$.

c. Use Stirling's approximation, equation (3.18), to show that $Q_k < e^k/k^k$.

$$Q_k = \left(\frac{1}{n}\right)^k \left(1-\frac{1}{n}\right)^{n-k} \binom{n}{k} < \left(\frac{1}{n}\right)^k \binom{n}{k} = \left(\frac{1}{n}\right)^k \frac{n!}{k!\,(n-k)!} = \frac{n!}{n^k(n-k)!}\frac{1}{k!}$$

$$< \frac{1}{k!} = \frac{1}{\sqrt{2\pi k}\left(\frac{k}{e}\right)^k} < \left(\frac{e}{k}\right)^k$$

$$\boxed{\frac{n!}{n^k(n-k)!} = \frac{n(n-1)(n-2)\ldots(n-k+1)}{n^k} < 1 \quad (for\ any\ k \ge 1)}$$

d. Show that there exists a constant $c > 1$ such that $Q_{k_0} < 1/n^3$ for $k_0 = c \lg n / \lg \lg n$. Conclude that $P_k < 1/n^2$ for $k \ge k_0 = c \lg n / \lg \lg n$.

$$Q_k \le \frac{e^k}{k!}$$

要证$Q_{k_0} < 1/n^3$只要证$\frac{e^{k_0}}{k_0!} < 1/n^3$对$k_0 = c\lg n/\lg\lg n$成立

$$\frac{e^{k_0}}{k_0!} < 1/n^3$$

$$\Rightarrow \frac{e^{c\lg n/\lg\lg n}}{(c\lg n/\lg\lg n)^{c\lg n/\lg\lg n}} < 1/n^3$$

$$\Rightarrow c\lg e \lg n/\lg\lg n - c\lg n/\lg\lg n * \lg(c\lg n/\lg\lg n) < -3$$

$$\Rightarrow \frac{c\lg e - c\lg c}{\lg\lg n} + c - \frac{c\lg\lg\lg n}{\lg\lg n} > 3$$

当$c > e$时

$$\Rightarrow c(1 - \frac{\lg\lg\lg n}{\lg\lg n}) > 3$$

又$\frac{\lg n}{n} \le \frac{1}{2}$

$$\Rightarrow c * \frac{1}{2} > 3$$

$$\Rightarrow c > 6$$

所以$k \ge k_0$时,$P_k \le P_{k_0} < 1/n^2$

$$
\begin{aligned}
E(M) &= \sum_{i=1}^{n} iP(M=i) \\
&= \sum_{i=1}^{c\lg n/\lg\lg n} iP(M=i) + \sum_{i=c\lg n/\lg\lg n}^{n} iP(M=i) \\
&\le (c\lg n/\lg\lg n) \sum_{i=1}^{c\lg n/\lg\lg n} P(M=i) + n * \sum_{i=c\lg n/\lg\lg n}^{n} P(M=i) \\
&\le (c\lg n/\lg\lg n)P(M \le c\lg n/\lg\lg n) + n * P(M > c\lg n/\lg\lg n) \\
&< (c\lg n/\lg\lg n) * 1 + n * P(M > c\lg n/\lg\lg n) \\
&< c\lg n/\lg\lg n + n * nP_k \\
&= \mathbf{O}(\lg n/\lg\lg n)
\end{aligned}
$$

e. Argue that

$$
\mathrm{E}[M] \le \Pr\left\{M > \frac{c\lg n}{\lg\lg n}\right\} \cdot n + \Pr\left\{M \le \frac{c\lg n}{\lg\lg n}\right\} \cdot \frac{c\lg n}{\lg\lg n} .
$$

Conclude that $\mathrm{E}[M] = O(\lg n/\lg\lg n)$.

CS 5.5- 8. Suppose you hash n items into k locations.

a. What is the probability that all n items hash to different locations?

b. What is the probability that the ith item is the first collision?

c. What is the expected number of items you hash until the first collision?

a: $P = \dfrac{\binom{k}{n} \times n!}{k^n}$ $(k \geq n)$

$P=0(k<n)$

$$\frac{k^{i-1} * (i-1)}{k^i}$$

The first i-1 items will harsh to different locations

B: $P = \dfrac{\binom{k}{i-1} \times (i-1)!}{k^{i-1}} \times \dfrac{i-1}{k} = \dfrac{\binom{k}{i-1} \times (i-1)! \times (i-1)}{k^i}$ $(k \geq i+1)$

$P=0(k<i+1)$

CS 5.5- 8. Suppose you hash n items into k locations.

a. What is the probability that all n items hash to different locations?

b. What is the probability that the ith item is the first collision?

c. What is the expected number of items you hash until the first collision?

$$P_i = \frac{k^{\underline{i-1}} * (i-1)}{k^i}$$

随机变量X表示冲突前被散列的关键字数

$$E(X) = \sum_{i=1}^{k} (i-1) * P_i = \sum_{i=1}^{k} (i-1) * \frac{k^{\underline{i-1}} * (i-1)}{k^i}$$

11. Suppose you hash n items into a hash table of size k. It is natural to ask how long it takes to find an item in the hash table. You can divide this into two cases, one in which the item is not in the hash table (an unsuccessful search) and one in which the item is in the hash table (a successful search). First consider the unsuccessful search. Assume the keys hashing to the same location are stored in a list, with the most recent arrival at the beginning of the list.

a. Using the expected list length, write a bound for the expected time for an unsuccessful search. Next, consider the successful search. Recall that when you insert items into a hash table, you typically insert them at the beginning of a list; thus, the time for a successful search for Item i should depend on how many entries were inserted after Item i.

b. Carefully compute the expected running time for a successful search. Assume that the item you are searching for is randomly chosen from among the items already in the table. (*Hint:* The unsuccessful search should take roughly twice as long as the successful one. Be sure to explain why this is the case.)

a

an unsuccessful search is to search all elements in a location

so the running time is 1 + n/k

B

A successful search is will search all elements which insert after the item we are searching

Then we define an indicator random variable $X_{ij}=\{h(k_i)=h(k_j)\}$

$Pr\{h(k_i)=h(k_j)\}=\frac{1}{k}$

$E(X_{i,j})= \frac{1}{k}$

So the running time is

$$E[\frac{1}{n}\sum_{i=1}^{n}(1+\sum_{j=i+1}^{n}X_{i,j})]=\frac{1}{n}\sum_{i=1}^{n}(1+\sum_{j=i+1}^{n}E[X_{i,j}])$$

$$=1+\frac{1}{nk}\sum_{i=1}^{n}(\mathrm{n}-\mathrm{i})=1+\frac{n}{2k}-\frac{1}{2k}$$