CS Chapter 5

5.5.8

a. If k<n, the probability is 0; If k≥n, the probability is $\frac{\binom{k}{n}\cdot n!}{k^n} = \frac{k^{\underline{n}}}{k^n}$

b. If the i^{th} element is the first to collide, then i-1 elements ahead of it mustn't collide. The number of that case is $\binom{k}{i-1}(i-1)!$, then the i^{th} element is bound to collide with one of them, there is $i-1$ cases in which that happens. So the total number of cases is $i \cdot \binom{k}{i-1}(i-1)! = \binom{k}{i-1}i!$, and the whole number of case of hashing i elements is $\binom{k}{i}i!$, so the probability is $\frac{\binom{k}{i-1}i!}{\binom{k}{i}i!} = \frac{i}{k-i+1}$

c. Assume random variable X denotes the elements number hashed when the first collision happens, then $Pr(X = i) = \frac{\binom{k}{i-1}i!}{\binom{k}{i}i!} = \frac{i}{k-i+1}$, so the expected number of elements hashed when the first collision happens $E(X) = \sum_{i=1}^{n} iPr(X = i) = \sum_{i=1}^{n} \frac{i^2}{k-i+1}$

d.

5.5.14

Define $X_i = I\{There\ is\ exactly\ i\ empty\ slots\}$,

$$X_i = \left((1-\frac{1}{k})^k\right)^i$$
$$= \left(1-\frac{1}{k}\right)^{ik}$$

Therefore, the expected number of empty slots is:

$$\sum_{i=0}^{2k-1} iX_i = \sum_{i=0}^{n-1} iPr(X_i)$$
$$= \sum_{i=0}^{2k-1} i\left(1-\frac{1}{k}\right)^{ik}$$
$$= \frac{(1-\frac{1}{k})\left[1-\left(1-\frac{1}{k}\right)^{(2k-2)k}\right]}{\left[1-\left(1-\frac{1}{k}\right)^k\right]^2} - \frac{\left(1-\frac{1}{k}\right)^k}{\left[\left(1-\frac{1}{k}\right)^k-1\right]} - \frac{(n-1)\left(1-\frac{1}{k}\right)^{nk}}{\left[\left(1-\frac{1}{k}\right)^k-1\right]}$$

Denote the equation above as *F(k)*,

$$\lim_{k\to\infty} F(k) = \frac{e}{(e-1)^2}$$

TC Chapter 11

11.2-3

It doesn't help to organize the elements in order, since there isn't an efficient algorithm to search a linked list. The time is still $O(1+\frac{n}{k})$.

11.2-6

We can view the hash table as a two-dimension array $H[m][L]$, in which there is mL grids, but they aren't not all full.

```
RANDOM-FIND(H, x)
do
    i = RANDOM(1,m)
    j = RANDOM(1,L)
while H[i][j] ≠ x
return i, j
```

11.3-3

Assume $x = \sum_{i=0}^{s} x_i (2^p)^i, y = \sum_{j=0}^{t} y_j (2^p)^j$, where s equals t. $X = \{x_1, x_2, \cdots, x_s\}$ is a permutation of $Y = \{y_1, y_2, \cdots, y_s\}$. As the hash function goes, $h(x) = \sum_{i=0}^{s} x_i (2^p)^i \bmod m = \sum_{i=0}^{s} x_i$, while $h(y) = \sum_{i=0}^{s} y_i (2^p)^i \bmod m = \sum_{i=0}^{s} y_i$. Hence $h(x) = h(y)$.

If we store a set of strings in a hash table, this situation is bad for looking for a certain pattern of permutation, since all permutations of the same string is in one slot.

11.3-4

$h(61) = 700$ $h(62) = 318$ $h(63) = 318$ $h(64) = 554$ $h(65) = 172$

11.4-2

Add a satellite data "*deleted*" to k:

```
HASH-DELETE(T, k)
i = 0
repeat
    j = H(k, i)
    if T[j] == k
        T[j].deleted = true
        return
    else
        i = i + 1
until i = m or T[j] = NIL
exit "Element not exist"
```

```
HASH-INSERT(T, k)
i = 0
repeat
    j = H(k, i)
    if T[j].deleted is true
        T[j] = k
        return j
    if T[j] == NIL
        T[j] = k
        return k
        i = i + 1
until i = m
exit "Hash table overflow"
```

11.4-3

Referred to theorem 11.6, when $\alpha = \frac{3}{4}$, the expected probe number is 4; when $\alpha = \frac{7}{8}$, it is 8.

11.1

a. Assume X is time of probing, then $Pr(X > k) = \sum_{i=k+1}^{n}\left(1 - \frac{i-1}{m}\right) \leq 2^{-k}\left(1 - \frac{1}{2^{n-k}}\right) \leq 2^{-k}$

b. *Proof.* Let $k = 2\log_2 n$, then $2^{-k} = 2^{-2\log_2 n} = \frac{1}{n^2}$, so the probability is $O(\frac{1}{n^2})$ □

c. *Proof.*

$$\begin{aligned} Pr(X > \log_2 n) &= Pr(X_1 > \log_2 n \cup X_2 > \log_2 n \cup \cdots \cup X_n > \log_2 n) \\ &= n \cdot O(\frac{1}{n^2}) \\ &= O(\frac{1}{n}) \end{aligned}$$

□

d. *Proof.*

$$\begin{aligned} E(X) &= \sum_{k=1}^{n} k \cdot Pr\{X = k\} \\ &= \sum_{k=1}^{2\log_2 n} k \cdot Pr\{X = k\} + \sum_{k=2\log_2 n}^{n} k \cdot Pr\{X = k\} \\ &\leq 2\log_2 n \cdot Pr\{X < k\} + n \cdot Pr\{X = 2\log_2 n\} \cdot (n - \log_2 n) \\ &\leq 2\log_2 n + n \cdot 2^{-2\log_2 n} \cdot n \\ &= 2\log_2 n + 1 \\ &= O(lgn) \end{aligned}$$

□

11.2

a. *Proof.* For a certain element x, the probability of it hashed to a certain slot is $\frac{1}{n}$, and it satisfies binary distribution. So the probability $Q_k = \left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k} \binom{n}{k}$ □

b. *Proof.*

$$\begin{aligned} P_k &= n \cdot Q_k \cdot Q_{<k}^{n-1} \\ &\leq n \cdot Q_k \end{aligned}$$

(Q_k means one slot has less than k elements, which happens exactly $n - 1$ times) □

c. *Proof.*

$$\begin{aligned} Q_k &= \left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k} \binom{n}{k} \\ &\leq \left(\frac{1}{n}\right)^k \binom{n}{k} \\ &= \frac{n!}{n^k \cdot k! \cdot (n-k)!} \\ &\leq \frac{1}{k!} \\ k! &= \sqrt{2\pi k}\left(\frac{k}{e}\right)\left(1 + \Theta(\frac{1}{k})\right) \\ &\geq \frac{e^k}{k^k} \\ Q_k &\leq \frac{1}{k!} \leq \frac{e^k}{k^k} \end{aligned}$$

□

d. *Proof.* Using the inequalities in c, let $k = k_0 = \frac{clgn}{lglgn}$. $\square$

e. *Proof.*

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

$\square$