

CS Problem 8:

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

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 $i \times \binom{k}{i-1}(i-1)! = \binom{k}{i}i!$, 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

$$E(X) = \sum_{i=1}^n i Pr(X = i) = \sum_{i=1}^n \frac{i^2}{k-i+1}$$

d.

Problem 11:

a. **Unsuccessful Search:** The expected length of a list is $\frac{n}{k}$, as the unsuccessful search will first calculate the address, then go through the desired list. In all, it takes $O(1) + O(\frac{n}{k}) = O(1 + \frac{n}{k})$ time.

Successful Search: Assume we search one of the n elements randomly, thus every element has a probability of $\frac{1}{n}$ to be selected. After calculating the hash function, we will go to the linked list to find the element (denoted by x). The times we need to check is the number of elements ahead of x , in other words, inserted after x plus one. We denote them as x_1, x_2, \dots, x_n , which means the i numbers (with $Max(i) = n$). Their key words are $k_1 = x_1.key, k_2 = x_2.key, \dots, k_i = x_i.key$. Define $X_{ij} = I\{h(k_1) = h(k_2)\}$, by the uniformity assumption, $Pr(X_{ij}) = \frac{1}{k}$. Hence, $E[X_{ij} = \frac{1}{k}]$. Then, time of a successful search is:

$$\begin{aligned} E\left[\frac{1}{n} \sum_{i=1}^n \left(1 + \sum_{j=i+1}^n X_{ij}\right)\right] &= \frac{1}{n} \sum_{i=1}^n \left(\sum_{j=i+1}^n E[X_{ij}]\right) \\ &= \frac{1}{n} \sum_{i=1}^n \left(1 + \sum_{j=i+1}^n \frac{1}{k}\right) \\ &= 1 + \frac{n-1}{2k} \\ &= 1 + \frac{n}{2k} - \frac{1}{2k} \\ &= O\left(1 + \frac{n}{k}\right) \end{aligned}$$

b.

Proof. As we can observe from a, the time for a successful search is approximately $\frac{n}{2k}$ (Ignore the time to calculate the hash function), while the time for an unsuccessful search is $\frac{n}{k}$ (Vice verse). So the unsuccessful search is roughly twice the time of a successful search. \square

Problem 14:

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

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

Therefore, the expected number of empty slots is:

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

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

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

TC 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 $O(1 + \frac{n}{k})$.

11.2-5

Since $|U| > mn$, so the expected length of the linked list is $\Omega(mn) \times \frac{1}{m} = \Omega(n)$. Hence there exists n elements hashed to one slot, search time for the worst case is $\Theta(n)$.

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

$i = \text{random}(1, m)$

$j = \text{random}(1, L)$

while $H[i][j] \neq x$

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 . $X = \{x_1, x_2, \dots, x_s\}$ is a permutation of $Y = \{y_1, y_2, \dots, y_s\}$. 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 $h(y) = \sum_{i=0}^s y_i(2^p)^i \bmod m = \sum_{i=0}^s y_i$. So $h(x) = h(y)$.

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 $T[j] == k$

$T[j].deleted = \text{true}$

return

else

$i = i + 1$

until $i = m$ or $T[j] = \text{NIL}$

exit "Element \square not \square exist"

```

HASH-INSERT(T, k)
i=0
repeat
    j=h(k, i)
    if T[j].deleted
        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 $Pr(X > k) = \sum_{i=k+1}^n (1 - \frac{i-1}{m}) \leq 2^{-k} (1 - \frac{1}{2^{n-k}}) \leq 2^{-k}$
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.

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

□

d.

Proof.

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

□

11.2

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

b.

Proof.

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

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

□

c.

Proof.

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

□

d.

Proof. Using the inequalities in c, let $k = k_0 = \frac{clgn}{lglg n}$.

□

e.

Proof.

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

□