Proof  Again we apply Markov's inequality (C.30), \( \Pr \{ X \geq t \} \leq \frac{\mathbb{E}[X]}{t} \), this time to inequality (11.7), with \( X = \sum_{j=0}^{m-1} m_j \) and \( t = 4n \):

\[
\Pr \left\{ \sum_{j=0}^{m-1} m_j \geq 4n \right\} \leq \frac{\mathbb{E} \left[ \sum_{j=0}^{m-1} m_j \right]}{4n} \\
< \frac{2n}{4n} \\
= \frac{1}{2}. \]

From Corollary 11.12, we see that if we test a few randomly chosen hash functions from the universal family, we will quickly find one that uses a reasonable amount of storage.

Exercises

11.5-1  *
Suppose that we insert \( n \) keys into a hash table of size \( m \) using open addressing and uniform hashing. Let \( p(n,m) \) be the probability that no collisions occur. Show that \( p(n,m) \leq e^{-n(n-1)/2m} \). (Hint: See equation (3.12).) Argue that when \( n \) exceeds \( \sqrt{m} \), the probability of avoiding collisions goes rapidly to zero.

Problems

11.1  Longest-probe bound for hashing
Suppose that we use an open-addressed hash table of size \( m \) to store \( n \leq m/2 \) items.

a. Assuming uniform hashing, show that for \( i = 1, \ldots, n \), the probability is at most \( 2^{-k} \) that the \( i \)th insertion requires strictly more than \( k \) probes.

b. Show that for \( i = 1, \ldots, n \), the probability is \( \Theta(1/n^2) \) that the \( i \)th insertion requires more than \( 2 \log n \) probes.

Let the random variable \( X_i \) denote the number of probes required by the \( i \)th insertion. You have shown in part (b) that \( \Pr \{ X_i > 2 \log n \} = O(1/n^2) \). Let the random variable \( X = \max_{1 \leq i \leq n} X_i \) denote the maximum number of probes required by any of the \( n \) insertions.

c. Show that \( \Pr \{ X > 2 \log n \} = O(1/n) \).

d. Show that the expected length \( \mathbb{E}[X] \) of the longest probe sequence is \( O(\log n) \).
Chapter 11  Hash Tables

11-4  Hashing and authentication

Let \( \mathcal{H} \) be a class of hash functions in which each hash function \( h \in \mathcal{H} \) maps the universe \( U \) of keys to \( \{0, 1, \ldots, m-1\} \). We say that \( \mathcal{H} \) is \( k \)-universal if, for every fixed sequence of \( k \) distinct keys \( \{x^{(1)}, x^{(2)}, \ldots, x^{(k)}\} \) and for any \( h \) chosen at random from \( \mathcal{H} \), the sequence \( \{h(x^{(1)}), h(x^{(2)}), \ldots, h(x^{(k)})\} \) is equally likely to be any of the \( m^k \) sequences of length \( k \) with elements drawn from \( \{0, 1, \ldots, m-1\} \).

\( a. \) Show that if the family \( \mathcal{H} \) of hash functions is 2-universal, then it is universal.

\( b. \) Suppose that the universe \( U \) is the set of \( n \)-tuples of values drawn from \( \mathbb{Z}_p = \{0, 1, \ldots, p-1\} \), where \( p \) is prime. Consider an element \( x = (x_0, x_1, \ldots, x_{n-1}) \in U \). For any \( n \)-tuple \( a = (a_0, a_1, \ldots, a_{n-1}) \in U \), define the hash function \( h_a \) by

\[
h_a(x) = \left( \sum_{j=0}^{n-1} a_j x_j \right) \mod p.
\]

Let \( \mathcal{H} = \{h_a\} \). Show that \( \mathcal{H} \) is universal, but not 2-universal. (Hint: Find a key for which all hash functions in \( \mathcal{H} \) produce the same value.)

\( c. \) Suppose that we modify \( \mathcal{H} \) slightly from part (b): for any \( a \in U \) and for any \( b \in \mathbb{Z}_p \), define

\[
h'_{ab}(x) = \left( \sum_{j=0}^{n-1} a_j x_j + b \right) \mod p
\]

and \( \mathcal{H}' = \{h'_{ab}\} \). Argue that \( \mathcal{H}' \) is 2-universal. (Hint: Consider fixed \( n \)-tuples \( x \in U \) and \( y \in U \), with \( x_i \neq y_i \) for some \( i \). What happens to \( h'_{ab}(x) \) and \( h'_{ab}(y) \) as \( a_i \) and \( b \) range over \( \mathbb{Z}_p \)?)

\( d. \) Suppose that Alice and Bob secretly agree on a hash function \( h \) from a 2-universal family \( \mathcal{H} \) of hash functions. Each \( h \in \mathcal{H} \) maps from a universe of keys \( U \) to \( \mathbb{Z}_p \), where \( p \) is prime. Later, Alice sends a message \( m \) to Bob over the Internet, where \( m \in U \). She authenticates this message to Bob by also sending an authentication tag \( t = h(m) \), and Bob checks that the pair \((m, t)\) he receives indeed satisfies \( t = h(m) \). Suppose that an adversary intercepts \((m, t)\) en route and tries to fool Bob by replacing the pair \((m, t)\) with a different pair \((m', t')\). Argue that the probability that the adversary succeeds in fooling Bob into accepting \((m', t')\) is at most \( 1/p \), no matter how much computing power the adversary has, and even if the adversary knows the family \( \mathcal{H} \) of hash functions used.