Universal hashing
Perfect Hashing

Arne Andersson
General Idea of Hashing

- Store elements (key, value) in an array
- Use the hash function to determine where each key is stored
- If the hash function is good, the keys are nicely spread
- If two keys have the same hash function, we have a collision, which must be handled
Handling collisions

Chaining

Open addressing

Double hashing:
Start at $h_1$
Jump $h_2$
How do we select a good hash function?
Universal Hash function

- A randomly selected hash function
- Works well with high probability for any set of keys
- Good example of randomized algorithm
Universal Hash function

- Given a set $\mathcal{H}$ of hash functions that maps keys into $0...m$.
- If for each pair $(x,y)$ the number of hash functions for which $h(x) = h(y)$ is at most $|\mathcal{H}| / m$, then $\mathcal{H}$ is universal.
Theorem 11.3

- If we store \( n \) keys into a table of size \( m \) using chaining, the expected length of the chain containing key \( k \) is \( n / m \) (\( = \alpha \))

- Proof sketch: for each other key, the probability of collision with \( k \) is \( 1 / m \)
Finding universal hash functions is easy!

- Class $H_{p,m}$ consists of all hash functions $h(k) = ((ak+b) \mod p) \mod m$

where

- $m$ is table size,
- $p$ is a prime, $p > m$
- $a$ and $b$ are random numbers

Theorem 11.5: $H_{p,m}$ is universal
With universal hashing, \textbf{expected} cost per operation is low.

But what if we want the \textbf{max} cost per operation to be low?

(Let’s say we wish to construct a static hash table to be stored on a CD-ROM and we want each search to be fast)
Perfect Hashing

Use secondary hash tables with quadratic size

\[ m_j = n_j^2 \]
Theorem 11.9
If we use a universal hash function to hash \( n \) keys into \( n^2 \) slots, the probability of any collisions is less than 1/2

\[
E \left[ \sum_{i=0}^{n-1} n_j^2 \right] =< 2n
\]
Theorem 11.9
If we use a universal hash function to hash \( n \) keys into \( n^2 \) slots, the probability of any collisions is less than \( 1/2 \)

Repeat trying: \( O(1) \) times before we find a hash function that gives no collision

Theorem 11.10
If we distribute \( n \) keys among \( n \) slots with a universal hash function, then

\[
E \left[ \sum_{j=0}^{n-1} n_j^2 \right] \leq 2n
\]

Total space & total construction time: \( O(n) \)
Perfect Hashing, Summary

Repeat
  Select the main universal hash function
Until \( \sum_{i=0}^{n-1} n_j^2 \leq 2n \)

For each slot
  Repeat
    Select the local universal hash function
  Until no collision

- \( O(n) \) space
- \( O(1) \) worst-case search cost
- \( O(n) \) expected construction cost