Hashing I

Andrew C. Lee

EECS, Syracuse

Contents

1. Reading: CLRS Chapter 11, Section 1 to 3. Drozdek, Chapter 10 and course notes from past semesters (CNPS).
2. Direct Address Table
3. Hash tables and Hash functions
4. Collision Resolution Strategies

Direct Address Tables

An *ideal* situation

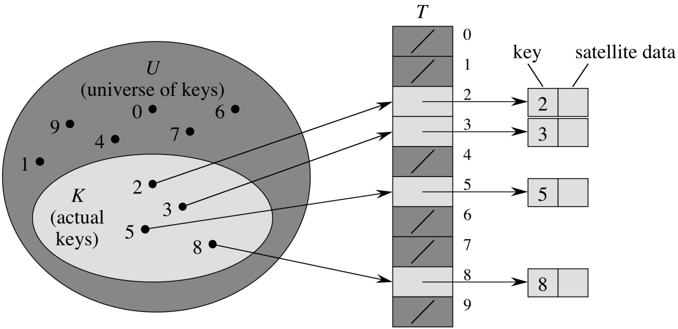

Figure : A Direct Address Table

Hash Tables

An *common* situation

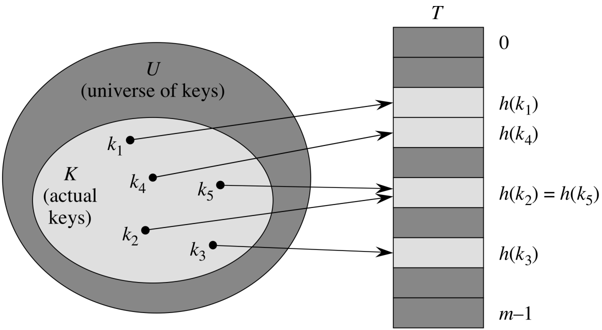

Figure : A Hash Table

Question What is h ? What is m ?

Example of Hash functions

1. Shift Folding
   Take parts of the key and add them together:
   A social security number (123-45-6789) can be divided into three parts and added
   $123 + 45 + 6789 = 6957$
   Then you can take the modulus of the table size
2. Boundry Folding
   The key is again divided into parts, but every other part is reversed
   $(123\text{-}45\text{-}6789) = 123 + 54 + 6789 = 6966$
   Again, then you can take the modulus of the table size
   Using bits that can actually reversing 456 is faster.

Example of Hash functions

1. Mid-Square Function
   Take the key, square it, and take the middle bits
   With this hash function in practice it is okay to have a power-of-two sized hashtable
2. Extraction
   Take only some of the bits or digits in a key
   Maybe all student id's start with 999, use the rest of the id as the key

See CNPS for more examples.

Collision Resolution Strategies

1. Separate Chaining
2. Open Addressing (elements occupy the table itself)
   2.1 Linear Probing
   2.2 Quadratic Probing
   2.3 Double Hashing
3. Coalesced hashing

Separate Chaining: Ideas

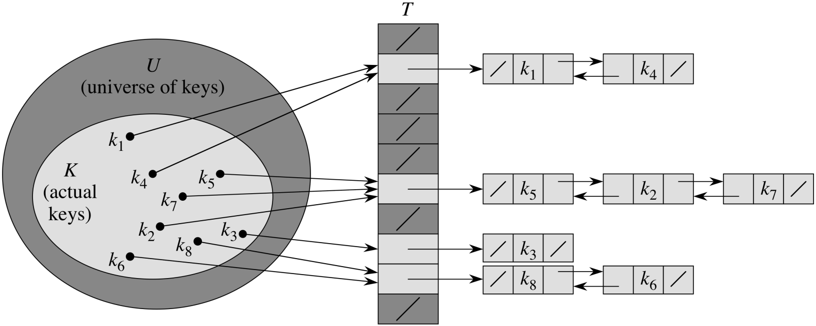

Figure : Build a Hash Table via Separate Chaining

Separate Chaining: Performance

1. Worst case is very bad (Why ?)
2. Interest: Average Case Performance
   Depends on how well the hash function h distributes the set of keys to be stored among the m slots, on the average.
3. When n (no. of keys) $= O(m)$ all dictionary operations is $O(1)$ run time *on average* (link lists used are double linked lists)

Open Addressing I

U: Universe of keys
m: Size of the hash table
h': Original hash function

1. The hash function is written as

$$h : U \times \{0, 1, \ldots, m-1\} \rightarrow \{0, 1, \ldots, m-1\}$$

   $h(k, i) =$ the address of i^{th}probe.
2. The probing sequence is $h(k, 0), \ldots, h(k, m-1)\}$.

Open Addressing II

1. Linear Probing: $h(k, i) = (h'(k) + i) \bmod m$
2. Quadratic Probing: $h(k, i) = (h'(k) + c_1 i + c_2 i^2) \bmod m$
3. Double Probing: $h(k, i) = (h_1(k) + i h_2(k)) \bmod m$

Examples

Refer to the examples from CNPS