Comp 352 Winter 2019

Tutorial Week 7

February 20, 2019

Outline

1. Map
2. Hash Table
3. Hash Functions – Collisions
4. Problem solving

Announcement

1. Mark
2. EAS

Map

- A Map is an ADT that allows us to store values based on unique keys.
- These entries are typically called key-value pairs.
- Multiple entries with the same key are not allowed.
- This is similar to how an array works except that instead of integer indexes, we use a more generic key as the "index".

Hash Table

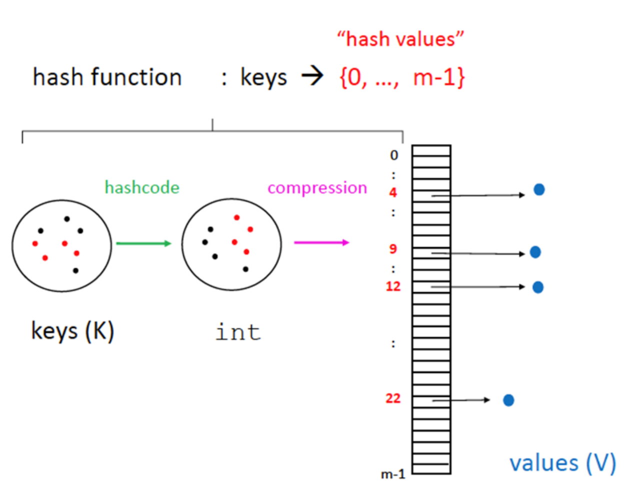
1. A hash table(hash map) stores data in the form of (key, value).
2. A hash table uses a hash function to compute an index into an array of buckets or slots, from which the desired value can be found.
3. It is often used to the situation that we need to fetch the desired key-value frequently.
4. Basic methods:
    - $get(k)$
    - $put(k, v)$ – add an entry with key k and value v
    - $remove(k)$ – remove the key-value entry e
    - $size()$
5. external: no children
6. leaves: external nodes

Hash Code and Hash Function

A hash function is usually specified as the composition of two functions:

1. Hash code:
   h_1: keys → integers
2. Compression function:
   h_2: integers → $[0, N-1]$
   - e.g.: h_2 = integers mod N
3. $h(x) = h_2\big(h_1(x)\big)$

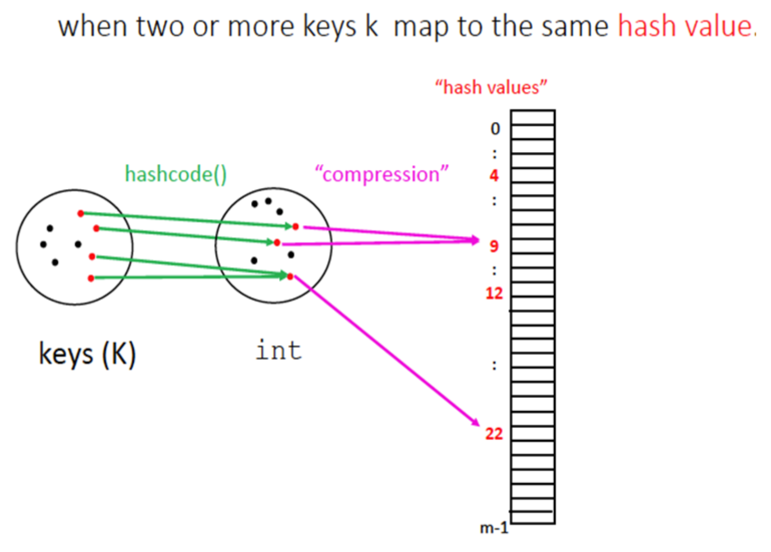
Hash Code and Hash Function



Hash Functions – Collisions

- One issue with hash tables is how well the Hash function behaves. That is to say, how well the keys map to integers.
- When two keys share the same hash value (result of the hash function), we get a collision.
- A good hash function minimizes collisions under most conditions.
- The way a hash table implementation handles collisions has an impact on the running time complexity of functions relying on the hash table.

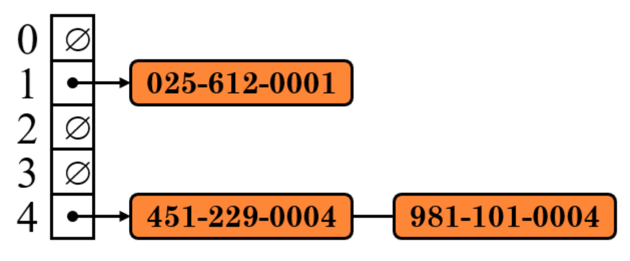
Collision



when two or more keys k map to the same hash value.

Collision Handling: Separate Chaining

The first strategy, commonly known as either open hashing, or separate chaining, is to keep a list of all elements that hash to the same value.



Collision Handling: Open Addressing

An alternative method to resolving collisions. The colliding item is placed in a different entry of the table instead of a linked list.

- Linear Probing:

  $$h_i(x) = \big(h(x) + i\big) \bmod N \text{ where } i = 0, 1, 2, \ldots$$

- Quadratic Probing:

  $$h_i(x) = \big(h(x) + i^2\big) \bmod N \text{ where } i = 0, 1, 2, \ldots$$

- Double Hashing

  $$h_i(x) = \big(h(x) + i \cdot h'(x)\big) \bmod N \text{ where } i = 0, 1, 2, \ldots$$

Problem solving

Assume an M entry hash table which needs to store N keys, where $h(i) = i \bmod M$.

What is the worst-case search time?

Problem solving

Consider the following hash table, where $h(x) = x\%9$.
What is the input?

Index	0	1	2	3	4	5	6	7	8	9
key	81	10	63			14	42			