CS2040/C

Friday, 12 October 2018

LAB DEMO 06

BBST

- map: store <key, value> pairs in particular order

  template<
    - class Key,
    - class T,
    - class Compare = std::less<Key>,
    - class Allocator = std::allocator<std::pair<const Key, T>>
    - > class map;
- Add elements: insert()
- Access elements: operator []
- Look up: count(), find()
- size()
- Iterators: begin(), end()

Hash Table

- unordered_map: store <key, value> pairs without any order

  template<

    - class Key,
    - class T,
    - class Hash = std::hash<Key>,
    - class KeyEqual = std::equal_to<Key>,
    - class Allocator = std::allocator<std::pair<const Key, T>>
    - > class unordered_map;

- Add elements: insert()
- Access elements: operator []
- Look up: count(), find()
- size()
- Iterators: begin(), end()

BBST

- set: store keys in a particular order

  template<

    - class Key,
    - class Compare = std::less<Key>,
    - class Allocator = std::allocator<Key>
    - > class set;

- Add elements: insert()
- Look up: count(), find()
- size()
- Iterators: begin(), end()

Hash Table

- unordered_set: store keys without any order

  template<

    - class Key,
    - class Hash = std::hash<Key>,
    - class KeyEqual = std::equal_to<Key>,
    - class Allocator = std::allocator<Key>
    - > class unordered_set;

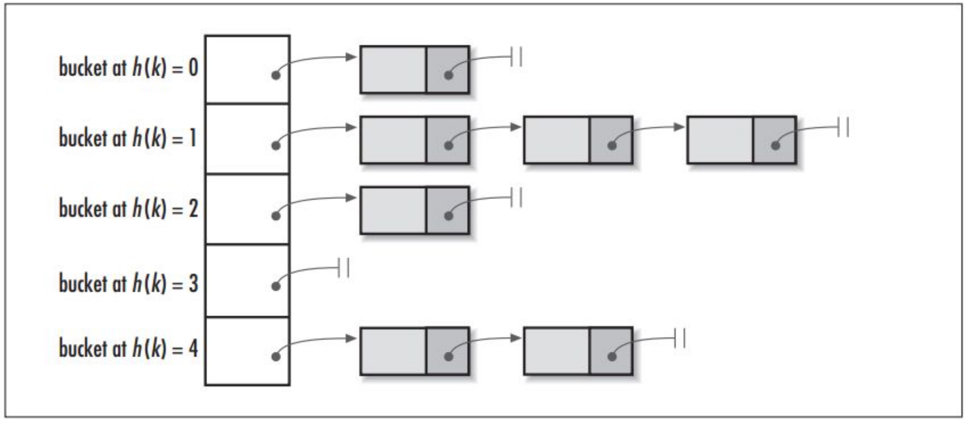
- Add elements: insert()
- Look up: count(), find()
- size()
- Iterators: begin(), end()

STL unordered/ordered set/map Question

- Write C++ code to create vocabulary from a text corpus
- Assume that words from the corpus is stored in a string array and given as input
- Use map, unordered_map, set, unordered_set for this task

Hash Table Question

- Create a dictionary to store students name and IDs with their age as the key
- Use direct addressing and separate chaining for this task


bucket at $h(k) = 0$
bucket at $h(k) = 1$
bucket at $h(k) = 2$
bucket at $h(k) = 3$
bucket at $h(k) = 4$