Assignment 11

Chained Hash Table

Grading

- Hash Table
  - 80%
  - Constructor
  - The `put` method
  - The `get` method
- Measurement and Analysis
  - 20%
  - Analysis of “unloaded” hash table (10x less than size of added items)
  - Analysis of loaded hash table (capacity 10x more than added items)

Hash Table Interface

- Hash211<K,V>
  - constructor: `Hash211<K,V>(int capacity, boolean printTimes)`
    - capacity - number of entries in table
    - printTimes - print times for `put` and `get` methods
- Methods
  - `V put(K key, V value)`
  - `V get(object key)`

Hash Table Implementation

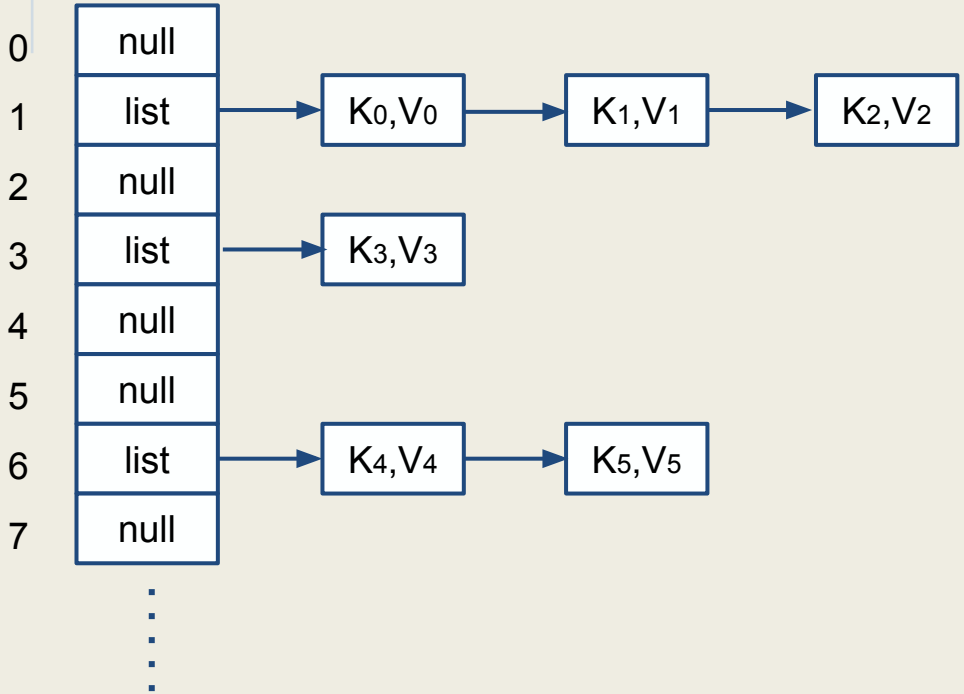
- Use `.equals(...)` to determine if two keys are the same
- Use `.hashCode()` of key to create index into table
  - Returns an `int`
  - Must make sure this value maps to a valid array location
- The `put` method
  - Adds new value to hash table
  - Returns the value just added
  - If a matching key already exists, old value is replaced
- The `get` method
  - Returns item matching key
  - Returns `null` if key does not exist in table

Hash Table

- Chained hash table
- Each element of hash table array is a linked list
- Elements of the hash table are linked lists
  - List elements contain both a key and a value
  - You can use the built-in java linked list class
  - Create a class for this (e.g.: `class KeyValuePair`)
  - `table[h] = new LinkedList<KeyValuePair>()`

```
class KeyValuePair<K,V> {
    K       key;
    V       value;
}
```

Chained Hash Table



Measurement and Analysis

- What's being measured
  - The amount of time a `put` operations take
  - The amount of time a `get` operations take
- Use nanoTime() to compute time deltas
  - This is required by the assignment
  - Time delta of zero - you'll need to use multiple `get` or `put` operations
- You can also use HashTableStressTest.java
  - Command line parameters
    - Specify capacity of hash table
    - Input word dictionary
  - See assignment for more details

Time Pseudo-Code

```
V put (K key, V value) {
    long      ts, tf;

    ts = System.nanoTime();

    // put implementation

    tf = System.nanoTime();
    System.out.println("put took " + tf-ts + "ns.");

    return val;
}
```

Example

- Example “dict.txt” has 234,937 elements
- Unloaded table would have capacity of 2,349,370 items
- Loaded table would have capacity of 23,493 items

Analysis Write-up

- Submit a text file, MS word doc, or PDF with your analysis
  - Just make sure it's included with your assignment submission
  - Can be in the jar file or separate
  - Just make sure you don't forget it
- Include your measurement results
- Write an analysis
  - Your measurements should be consistent with expected hash table performance
  - Your book has a good description of hash table performance
  - Your measurements should support the analysis you write

Bonus Point

- Add comments
- Create readable code
- Make your code efficient
- Etc…

...and get extra credit!