CSCI 102 assignment 4 – Hash maps

March 4, 2024

1 Separate chaining

In this assignment you will be implementing a hash map that resolves collisions by chaining.

- Define a class `HashMapChain<K, V>` that implements `Map<K, V>`. It will have three private attributes, the size `int size`; the list of chains `UnsortedMap<K, V> maplist[]`; and a variable that describes the number of chains `static int maxSize`.

- Create two constructors – one that takes in `int maxSize` and another default constructor that sets `maxSize` to 1000.

- To get the hash code of a `key K`, call `k.hashCode()`; this returns the default java hash code.

- Override `hashCode` in `DoublyLinkedList` to get a hash function that turns the elements in the list into a string and then hashes the concatenated string with commas in between and brackets on the ends– i.e. a list that has the doubles `3.43` and `5.432` should have the hash of the string "[3.43,5.432]". To get the string of an element, use the built in Object method `toString()`.

- Build a `main` method in `HashMapChain` where you create a hash map of doubly linked lists of doubles with keys that are `String`s. Create and add the lists [3.43,5.432], [7], [812.4, 12.76, 123.4] and add them with keys "One", "Two", "Three". Print out the result of `get(''Two'')`. Try to add [1, 2, 3.4, 4, 5] with key "Two" and print the result of `get(''Two'')` again.

2 Open addressing

Don't worry, I won't make you implement open addressing. Please answer the following question, similar to question 10.7 in the textbook. Imagine a hash table with 8 elements that takes entries with `int` keys and values and hash function $h(i) = i \mod 8$. Draw the table after each of the operations below and write what is returned in each case, resolving collisions with linear probing. Indicate which cells are visited during each operation. Do not forget to draw a placeholder entry if necessary.

`put(0, 3)`, `put(3, 17)`, `put(1, 12)`, `put(8, 8)`, `put(5, 3)`, `remove(1)`, `put(7, 1)`, `get(8)`, `put(16, 2)`, `put(15, 2)`, `put(24, 9)`, `remove(3)`, `remove(16)`, `remove(8)`, `get(24)`.

Please submit your code and answers to the questions in a zipped folder on Brightspace by March 11.