CS 1332 Recitation 6 Worksheet – Hash Maps

CS 1332 TAs

February 18, 2019

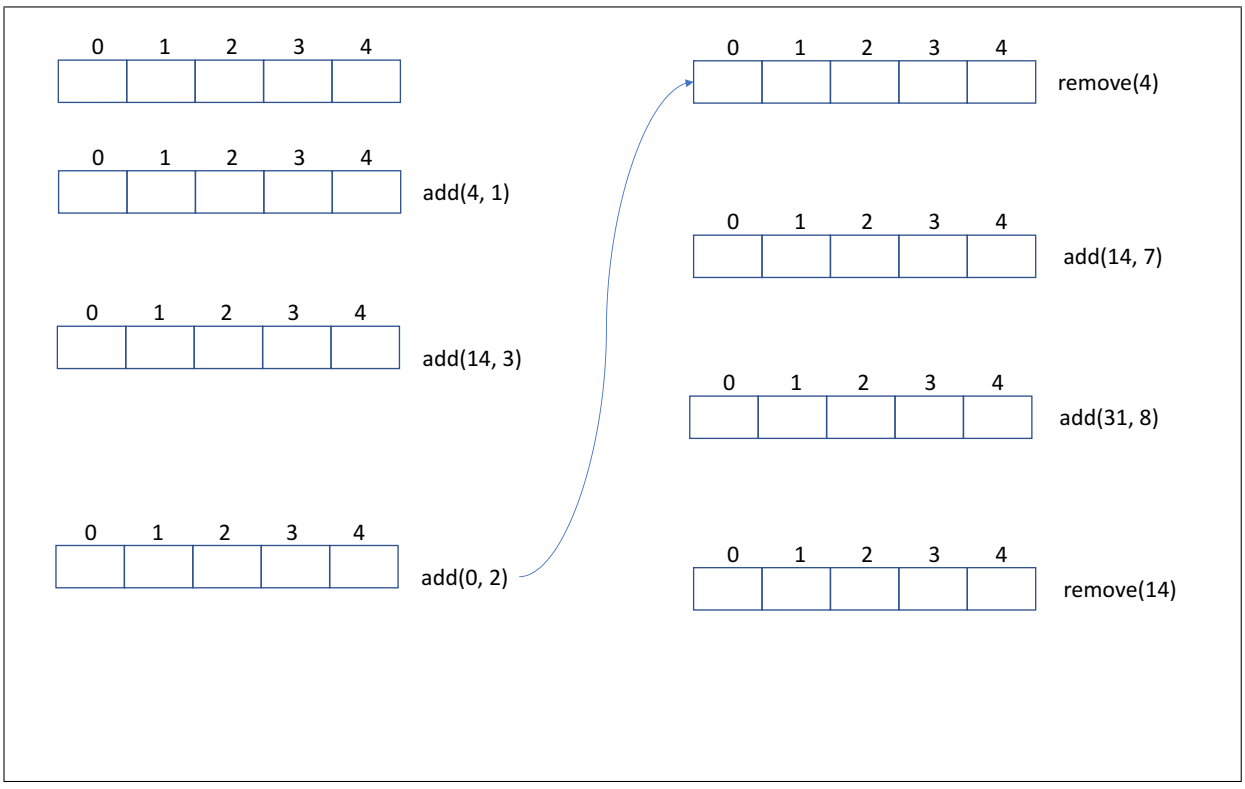
This worksheet covers material from this week's recitation. It is meant to be additional exercise for your benefit and will not be graded. Feel free to collaborate with other students on this and ask TAs for help. Distribution of this worksheet is **not** permitted.

1 Questions

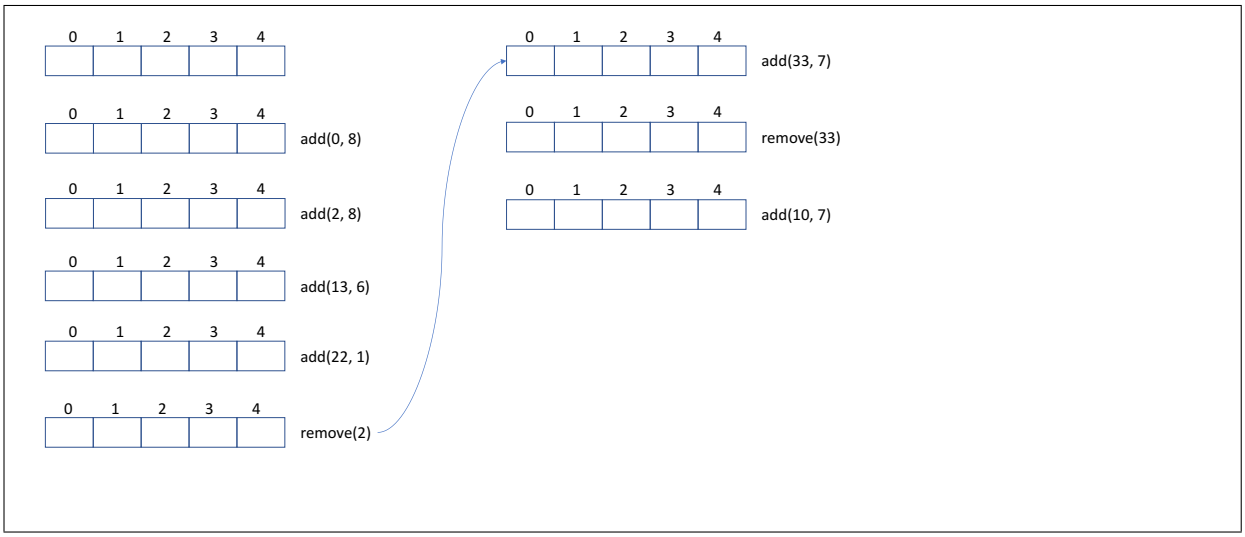
1. Runtime of adding to a hash map with external chaining and a good hash function: _____
2. Runtime of adding to a hash map with linear probing and a hash function that always returns 4: _____
3. Runtime of adding to a hash map with quadratic probing and a hash function that always returns 4: _____
4. Average-case of adding to a hash map with external chaining backed by BSTs instead of linked lists: _____
5. Worst-case of adding to a hash map with external chaining backed by BSTs instead of linked lists: _____
6. Why can't we use BSTs in place of linked lists for hash maps in general?: _____

2 Diagramming

1. Perform the following operations on a hash map with external chaining. Add to the back of the linked list when necessary. The hash function on keys is just the key itself. Ignore load factor and resizing



2. Perform the following operations on the hash map with linear probing. The hash function on keys is just the key itself. Use "DEL" to specify deleted entries. Ignore load factor and resizing



3 Above and Beyond

This is a small set of challenge questions related to the material in the course, similar to those that you may find in a Software Engineering interview. Again, feel free to collaborate with other students or ask a TA for help.

1. Given an arbitrary array and an int c, write a `threeSum` method that returns true if the array has three values that sum to c.

   Examples:
   Input: $[2, 1, 8, 3, 5]$, $c = 12$ Output: true

   Expected time: $O(n^2)$, Expected space: $O(n)$ outside of original array

```
public boolean twoSum(int[] arr, int c) {

}
```

2. Write a function `lswrc()` that returns the longest substring without repeating characters in a given `String`.

   Example:
   Input: pwwkew
   Output: 3
   Expected time: $O(n \log k)$, Expected space: $O(k)$ outside of original array.

```
public int lswrc(String s) {

}
```