CS61B Spring 2016 Secret Section 5 Worksheet

4/4

1 Heap Refrehser

(a) Give the array that results if we use transform the array [9,3,7,2,1,4,5] into a binary max heap.

(b) Remove 9. What does the heap look like now? What is the runtime for removal?

(c) Insert 10. What does the heap look like now? What is the runtime for insertion?

(d) Check if the heap contains 6. What is the runtime for checking if a heap contains an element?

2 Bored of Hashing?

(a) Suppose we are keeping a HashSet of 8-puzzle boards for some homework in some Berkeley CS class. The boards use a hashcode that takes the tile values of the upper left tile and bottom right tiles (0 for the blank space) and returns their sum. *Assume that the boards are evenly distributed by this hashcode.* Using **N** = the number of boards in the HashSet, what is the worst case big-O time for adding a new board? See if you can derive the worst case tilde notation time as well.

(b) Now suppose we discover our HashSet needs to concurrently store 8-puzzle boards of varying sizes (So there might be 1000×1000 boards alongside 3×3 boards). You can assume the board sizes in the HashSet are evenly distributed. Using the same hashcode and assumptions from part (a), how does the worst-case big-O time change?

3 Yelplet

In this problem we will be making a class for a microscopic version of Yelp! This version only keeps track of business and the days that they are open.

To satisfy users, try to make `isBusinessOpen()` and `getBusinessesByDay()` as efficient as possible (imagine when a lot of businesses have been added already).

Assumptions you can make:

- All argument inputs will be sane (eg. non-null, and inputs to the day argument will always be a valid day)
- You have a helper method `int dayToIndex(String day)` which takes in ”Monday” as a string and returns 0, ”Tuesday” as a string and returns 1, etc. to ”Sunday” returning 6.

```
public class Yelplet {
    /* Static or instance variables here */



    public Yelplet() {






    }
```

```java
static int dayToIndex(String day) { ... }

/* Stores a business by its name and the days it is open */
public void addBusiness(String name, List<String> daysOpen) {







}

/* Returns true if a business is open on the given day */
public boolean isBusinessOpen(String name, String day) {







}


/* For a specific day, returns the business that are open on that day in no specific order */
public List<String> getBusinessesByDay(String day) {







}
```

4 Huffman Coding

Huffman coding is a technique used for converting words to short codes when compressing data. The input is typically given as a list of words and frequencies:

Swag: 5
Best: 6
Is: 7
CS61B: 15
The: 6

Let's say each word-frequency pair is a node. When running Huffman coding, you first find the two nodes A, B with the least frequencies. You then create a new node C with children A, B, whose frequency is the sum of the two. You repeat this process until all the nodes are connected.
(Note: The newly-created node C contains a frequency and should be included when searching for the least frequent nodes. The newly-connected nodes A, B are removed from the search since they are now combined under C)

What data structure(s) would you use to run Huffman coding?

Run Huffman coding using these data structure(s) until the final state.

Let's say we want to use your data structure, in its final state, to find words with frequencies in the range $6 \leq f \leq 7$. With that constraint, use an arbitrary traversal/search and give the search's resulting word order.