19.7 Exercises

1. Potpourri

(a) True or False: Resizes are triggered if adding a key-value pair causes the load factor to be greater than or equal to the specified maximum load factor.

Answer to Q1(a)

False. A resize will be triggered if adding another key value pair would cause the load factor to **exceed** the specified maximum load factor. A resize is not triggered when the load factor is equal to the specified maximum load factor.

(b) True or False: The `hashCode()` function can have varied return types.

Answer to Q1(b)

False. In Java, the hashCode() function can **only** have an `int` return type to serve as a hash value.

2. hashCode()....

In order for a hash code to be valid, objects that are equivalent to each other (i.e. .equals() returns true) must return equivalent hash codes. If an object does not explicitly override the `hashCode()` method, it will inherit the `hashCode()` method defined in the Object class, which returns the object's address in memory.

Here are four potential implementations of Integer's `hashCode()` function. Assume that `intValue()` returns the value represented by the Integer object.

Categorize each `hashCode()` implementation as either a valid or an invalid hash function. If it is valid, point out a flaw or disadvantage. If it is invalid, explain why.

Question (a)

```
public int hashCode() {
    return -1;
}
```

› Answer for Q2(a)

Question (b)

```
public int hashCode() {
    return intValue() * intValue();
}
```

› Answer for Q2(b)

Question (c)

```
public int hashCode() {
    Random rand = new Random();
    return rand.nextInt();
}
```

› Answer for Q2(c)

Question (d)

```java
public int hashCode() {
    return super.hashCode();
}
```

› Answer for Q2(d)

3. Hashin' and Resizin'

Given the provided `hashCode()` implementation, hash the items listed below with external chaining (the first item is already inserted for you). Assume the load factor is 1, and the initial underlying array has size of 4. Use geometric resizing with a resize factor of 2. You may draw more boxes to extend the array when you need to resize.

```java
/** Returns 0 if word begins with ’a’, 1 if it begins with ’b’, etc. */
public int hashCode() {
    return word.charAt(0) - 'a';
}
```

```
["apple", "cherry", "fig", "guava", "durian", "apricot", "banana"]
```

› Answer for Question 3

Previous
19.6 Summary

Next
20. Hashing II