CS 61B Spring 2017

Hashing

Discussion 9: March 15, 2017

1 Data Structures

Give a tight asymptotic bound for each of the following problems. Provide your bound in $\Theta(\cdot)$ if it exists, otherwise provide both the $O(\cdot)$ and $\Omega(\cdot)$ bound.

1.1 Insertion of one element into each of the following data structures where N is the number of elements already in the collection.

(a) `ArrayList`

(b) `LinkedList`

(c) `BSTMap` (binary search tree)

(d) `TreeSet` (balanced search tree)

(e) `HashSet`

1.2 Containment check (`contains`) of one element in each of the following data structures where N is the number of elements already in the collection.

(a) `ArrayList`

(b) `LinkedList`

(c) `BSTMap` (binary search tree)

(d) `TreeSet` (balanced search tree)

(e) `HashSet`

1.3 Suppose we're designing a hash table. Compare and contrast each of the following external chaining implementations. Why would you use one over the other?

(a) Linked list

(b) Resizing array

(c) Balanced search tree

(d) Hash table

2 Hash Codes

There is a problem with each `hashCode()` method below (correctness, distribution, efficiency). Assume there are no problems with the correctness of `equals()`.

2.1
```
class PokeTime {
    int startTime;
    int duration;
    public int getCurrentTime() {
        // Gets the current system clock time
    }
    public int hashCode() {
        return 1021 * (startTime + 1021 * duration + getCurrentTime());
    }
    public boolean equals(Object o) {
        PokeTime p = (PokeTime) o;
        return p.startTime == startTime && p.duration == duration;
    }
}
```

2.2
```
class Phonebook {
    List<Human> humans;
    public int hashCode() {
        int h = 0;
        for (Human human : humans) {
            // Assume Human::hashcode is correct
            h = (h + human.hashCode()) % 509;
        }
        return h;
    }
    public boolean equals(Object o) {
        Phonebook p = (Phonebook) o;
        return p.humans.equals(humans);
    }
}
```

2.3
```
class Person {
    Long id;
    String name;
    Integer age;
    public int hashCode() {
        return id.hashCode() + name.hashCode() + age.hashCode();
    }
    public boolean equals(Object o) {
        Person p = (Person) o;
        return p.id == id;
    }
}
```