Discussion 11
Prelim 2 review

CS 2110, SP24

Topics

- Recursion
- Trees
- Loop Invariants & searching
- Sorting & comparisons
- Dictionaries & hashing
- GUIs & lambda expressions
- Concurrency

Recursion

Sequence of Recursive Calls

```java
/** returns the nth fibonacci number,
  * where fib(0) and fib(1) are 1
  * Precondition: n must be positive
  * */
public static int fib(int n){
      if (n==0 || n == 1) {
          return 1;
      } else {
          return fib(n-1) + fib(n-2);
      }
}
```

What is the sequence of calls for:

`fib(0)`

`fib(2)`

`fib(4)`

For each call, write the output returned.

Implementing a recursive function: `tacoCat()`

Recursively implement the following method according to its specification:

```
/**
  * Returns the tacocat-version of input String [s]. A tacocat-version of
  * String [s] is [s] concatenated with the reverse of [s], but without the
  * resulting middle character duplicated.
  *
  * Ex: tacoCat("taco") ⇒ "tacocat"
  *     tacoCat("race") ⇒ "racecar"
  *     tacoCat("kay")  ⇒ "kayak"
  */
public static String tacoCat(String s) {

    // Your implementation here…

}
```

Analyzing algorithm

Consider some BST class, and assume it has method `add()`, which adds nodes to the tree in BST format.

Consider the search algorithm shown. What is the best and worst time complexities of this method in order of the size of the tree?

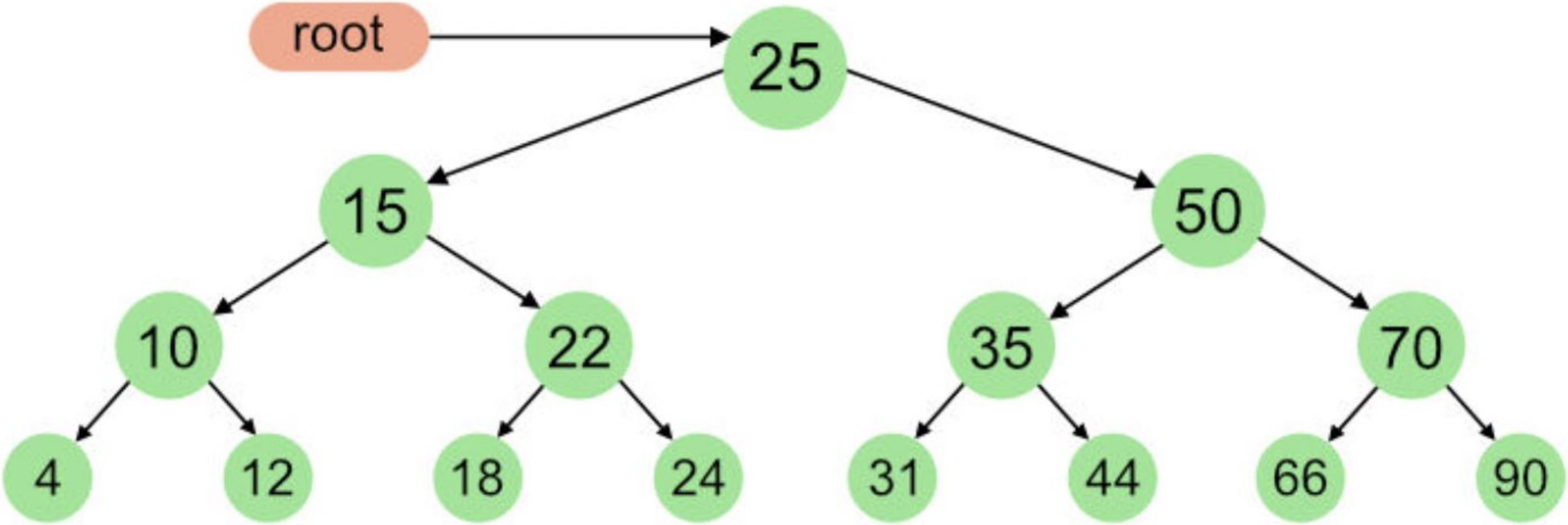
```java
private boolean searchRecursively(TreeNode root, int value) {
    if (root == null) {
        return false;
    }

    if (root.value == value) {
        return true;
    }

    if (value < root.value) {
        return searchRecursively(root.left, value);
    } else {
        return searchRecursively(root.right, value);
    }
}
```

Trees

Give a preorder, inorder and postorder traversal of this tree:



Example taken from: https://www.geeksforgeeks.org/tree-traversals-inorder-preorder-and-postorder/#

Big O notation

What is the worst-case time complexity for the following, given a BST of *n* nodes?

1. Checking if a number is in the BST

2. Computing the depth of the BST

Binary Trees and Binary Search Trees

```
        5
       / \
      /   \
     4     10
    / \   / \
   3  17 8   21
```

```
         50
        /  \
       /    \
      25     75
     /  \   /  \
    20  30 70   80
```

For both of these trees, state whether each of them are a:

- Binary Tree
- Binary Search Tree

Loop invariants & searching

State the numbers that are contained in the sequences below, given the following range notation:

[1,2]

[1,2)

(2,2)

(1,4]

a[1..4]

a[....4)

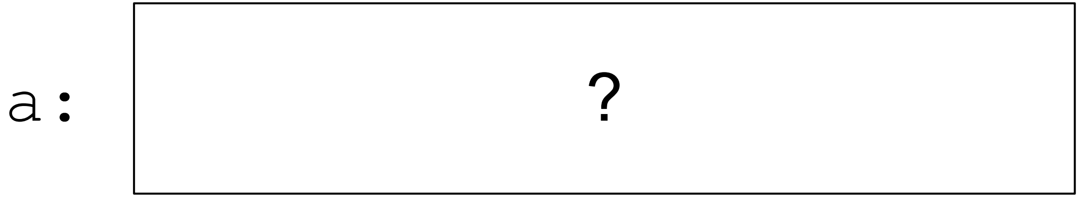
a[....4]

a(1…]

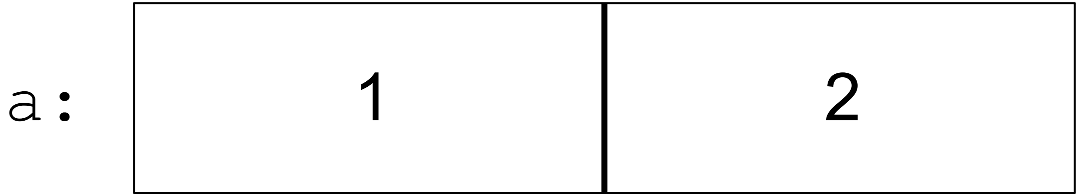
Creating Loop Invariants from Preconditions and Postconditions

The array `a`, declared as `int[] a`, contains elements that are either `1` or `2`. Given the precondition and postcondition shown below, draw a possible loop invariant for this array.

Precondition

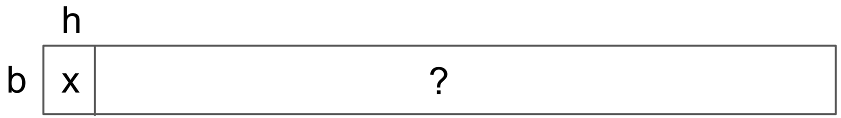


Postcondition

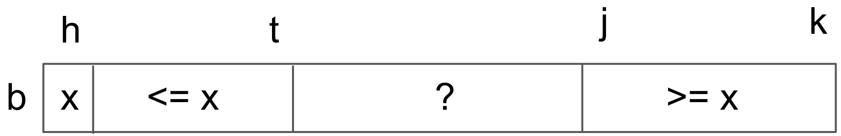


Fill in the blanks to complete the loop:

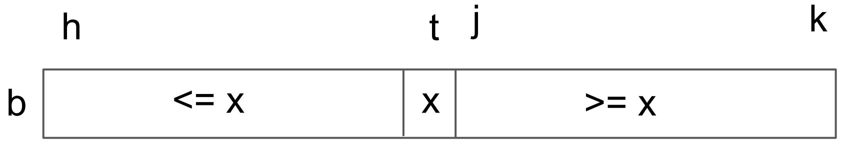
Precondition:



Loop Invariant:



Postcondition (**after** post-loop swap):



Fill in the missing parts (1-8) of the following code to make the loop correct :

```
public static void partition(int[] b, int h, int k){
    int t = (1)________;
    int j = (2)________;
    int x = b[h];

    while(t < (3)______){
        if (b[(4)_______] <= x){
            t++;
        } else {
            swap(b, (5)____, (6)____);
            j--;
        }
    }
    swap(b, (7)____, (8)____);
}
```

Note: this is the *method* postcondition, not the *loop* postcondition (it is not consistent with the invariant).

Sorting & comparisons

Complete the following nested `DescendingComparator` class to sort `Student` objects in order of **decreasing** birth year:

```
public class Student {
    private String name;
    private int yearOfBirth;
    …
    /* constructor and observers */
    …
    public class DescendingComparator implements Comparator<Student> {
        // TODO: implement compare() method
    }
}
```

Author: Karen Sabile (ks2238)

State the worst-case *time complexities* (in terms of the number of items to be sorted, N) for:

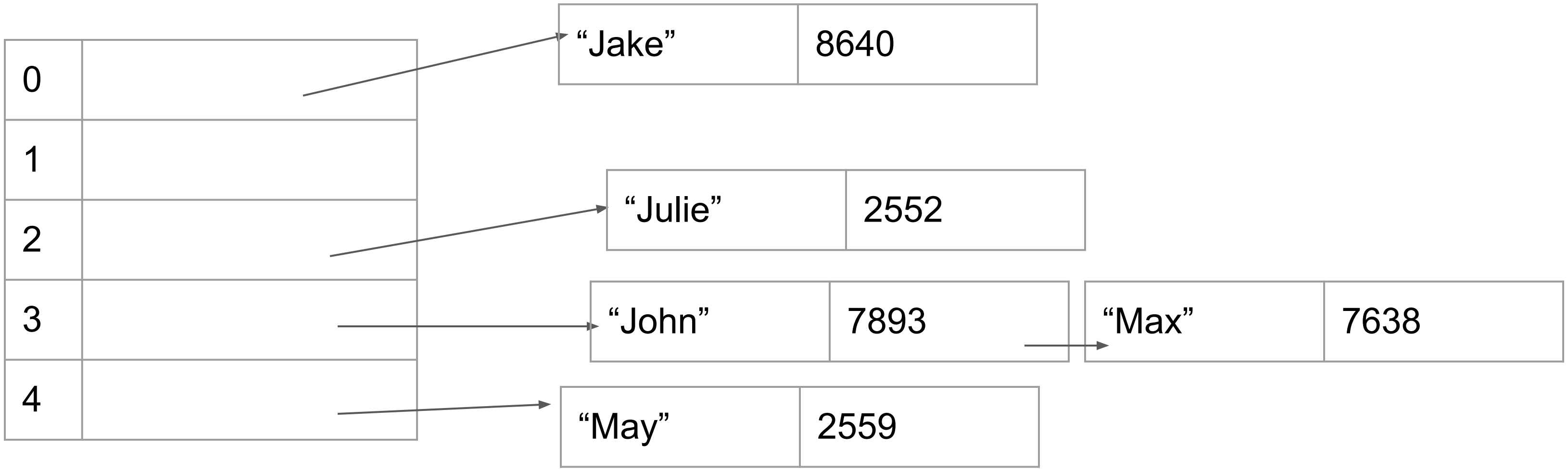
- Selection Sort
- Insertion Sort
- Merge Sort
- Quick Sort

What is the worst case *space complexity* for the following, in terms of the number of items to be sorted, N?

- Selection Sort
- Insertion Sort
- Merge Sort
- Quick Sort

Dictionaries & hashing

Draw the following hashtable (representing a Set) after it is resized to have a length of 9. The names are the keys, and the values are the hashes associated with each key.



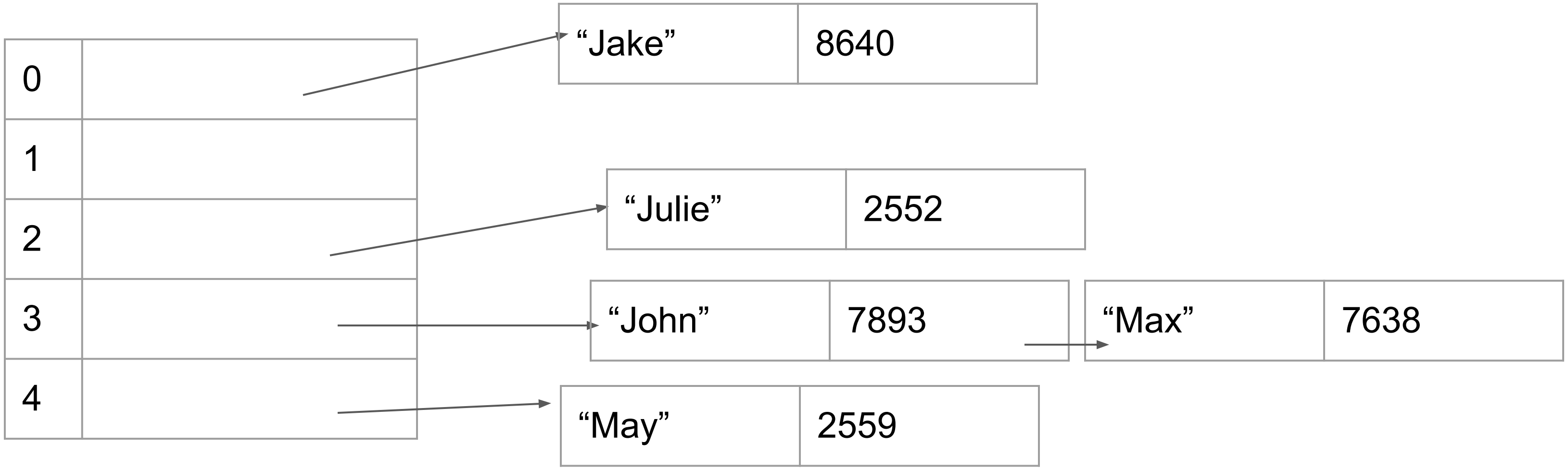
Below is a hash table that uses linear probing. We want to insert an entry *e* whose key has hash code of 26. What index would *e* belong in?

0	1	2	3	4	5	6	7	8	9
10		2				36	17		29

Load Factors

Compute the load factor of the hash table below.

What should the new bucket size be if we want the load factor to be <= 0.75?



GUIs & lambda expressions

Identify the event object, event source, and listener for handling button clicks.

```
class App extends JFrame
   implements ActionListener {
  App() {
    JButton b = new JButton("B");
    add(b);
    b.addActionListener(this);
  }
  @Override
  public void actionPerformed(
   ActionEvent e) {
    print("Prelim " + e.getSource());
  }
}
```

Event Dispatch Thread (EDT)

An operation needed for a program takes 10 minutes to run. Would it be appropriate to execute the operation on Swing’s event dispatch thread?

Lambda Expressions

Write a lambda expression to make button b print “Big Red Bear” when it is clicked.

```
class App extends JFrame {
  App() {
    JButton b = new JButton("B");
    add(b);
    b.addActionListener(???);
  }
}
```

Asserting Exceptions

Write an assertion using **`assertThrows()`** to assert that dividing an integer by 0 will throw an ArithmeticException.

Hint: One of the arguments to **`assertThrows()`** is a lambda expression.

Concurrency

Three cats are at a table. Each cat cannot start eating until they have two chopsticks. They acquire one chopstick at a time and do not put them down until they have eaten. (The cats know how to use chopsticks and they refuse to eat without them.)

a) If there are 4 chopsticks, is it possible for all of the cats to starve?
b) If there are 3 chopsticks, is it possible for all of the cats to starve?

What computer science term describes the situation in which all cats starve?

Execution of Two Threads

Assume that the initial value of z is 3 and consider the code below in two threads. What are the possible values of z after execution of the two threads?

Thread 1:

z = z * 2

Thread 2:

z = z + 5

Deadlock

Given two sequences of instructions (in pseudocode or Java) that are executed concurrently and require exclusive access to one or more shared resources, state whether deadlock is possible. If so, propose an alteration to one of the sequences that eliminates this possibility while preserving the desired behavior of the original instructions.

Process 1:

1) Acquire X
2) Use X
3) Release X
4) Acquire Y
5) Acquire Z
6) Use Y and Z
7) Release Y
8) Release Z

Process 2:

1) Acquire X
2) Use X
3) Release X
4) Acquire Z
5) Acquire Y
6) Use Y and Z
7) Release Y
8) Release Z

Given some variables X, Y, and Z on these two processes:

- Can deadlock occur if these processes happen simultaneously?
- If so, where, and what change can be made to avoid deadlock?