CS 125 - Lecture 36

Objectives: recursive trees; binary search; recursive music;

Up next: MP6 due today (8PM);

1. **Review Selection Sort:** What helper methods are used?

Recursive case:

Base case:

2. Write a function to reverse print a list of words. The following java program will read and print the word list in original order:

```
public class PrintWords {
   public static void main(String[] args) {
      TextIO.readFile("/Users/chapman/Documents/workspace/
PrintWords/src/original.list.text");
      // while (! TextIO.eof()) {
       //  String line = TextIO.getln();
       //  TextIO.putln(line);
      // }
      reversePrint();
   }
   public static void reversePrint() {









   }
}
```

3. **Discuss:** How do you use a phone book to search for a phone number?

Is this faster than linear search?

Are there limitations to searching this way?

4. **Binary Search**:

Example use: `int indexFound = binSearch(data, 0,data.length-1,23.0);`

```
//  return -1 if the target value is not found in the array

public static int binSearch(double[] array,
                   int lo, int hi, double target) {

     int mid = (lo + hi) / 2;









}
```

How would you invoke this method?

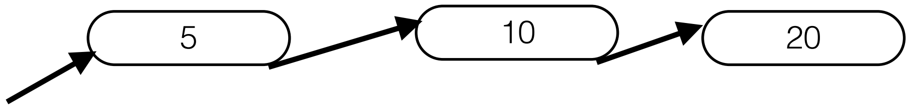
Is there a better way?

6. **If each link has a larger value than the previous, will the following** `getMax()` **create a tree or chain of activations?**

```
class LinkedList{
   int value;
   LinkedList next;
}
public int getMax() {
   if (next == null)
      return value;  // BASE CASE

   int result = next.getMax();
   if (result < value) return value;
   else return next.getMax();
}
```

7. Write a recursive song in ABA format:

```
//  Recursive method to create the pitches for son
//  Assume pitches in array have all been initialized to 440.0;

public static void createSong(double[] pitches, int lo, int hi, double
augment) {

   // divide the range of subarray into thirds and work on each third

   int oneThird = (hi - lo + 1) / 3;








}
```

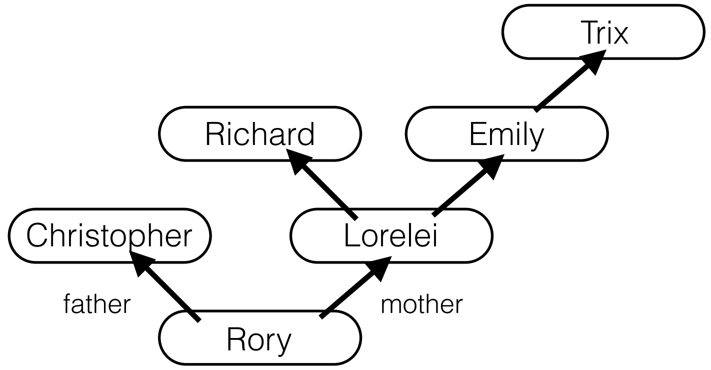


6. How can we insert links to create a *sorted* linked list?

```
     list = new LinkedList(10,null);
     list = list.insert(20);
     list = list.insert(5);
     public LinkedList(int newValue, LinkedList newNext)
                                         { … } //constructor
```

Write a function that takes an int and inserts in order:

```
public LinkedList insert(LinkedList list, int value) {
    if (list == null || value < list.value)
      return new LinkedList(value, list);
    else {
      list.next = insert(list.next, value);
      return list;
    }
}
```



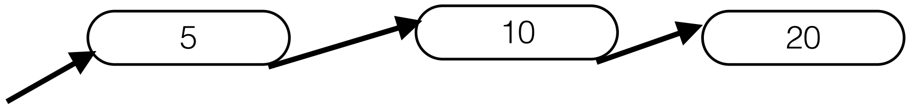
8. Write a recursive method to return the total number of people in this family tree; use accumulator argument:

6. Write a function to reverse print a list of words. The following java program will read and print the word list in original order:

```
public class PrintWords {
   public static void main(String[] args) {
      TextIO.readFile(“/Users/chapman/Documents/workspace/PrintWords/
src/original.list.text”);
      // while (! TextIO.eof()) {
       //  String line = TextIO.getln();
       //  TextIO.putln(line);
      // }
      reversePrint();
   }
   public static void reversePrint() {









   }
}
```

8. You have an array of doubles. You want to search between indices 'lo' and ‘hi'. Write a recursive method to find the largest product of two neighboring values. e.g. `findPair({ 1.0 , 1.0 , 7.5 , 4.0, 4.1 , 3.5 },0,5)` returns 30.0 (7.5 * 4.0), which is largest product of two neighboring values.

```
public static double findPair(double[] array, int lo, int hi) {









}
```



5. How can we insert links to create a *sorted* linked list?

```
     list = new LinkedList(10,null);
     list = list.insert(20);
     list = list.insert(5);
     public LinkedList(int newValue){ … } //constructor
```

Write a function that takes an int and inserts in order:

```
public LinkedList insert(LinkedList list, int value) {
    if (list == null || value < list.value)
      return new LinkedList(value, list);
    else {
      list.next = insert(list.next, value);
      return list;
    }
}
```

7. Given array Contact[], containing names and phone numbers sorted by name, write an efficient search algorithm. Return phone number for name.

```
public class Contact {
   public int phoneNumber; public String name;
}
```

e.g. `int number = find(friends, “Larry Page”, 0, friends.length-1);`

```
public static int find(contacts[] array, String friend,
                                        int lo, int hi) {









}
```

5. You have an array of doubles. You want to search between indices 'lo' and ‘hi'. Write a recursive method to find the largest product of two neighboring values. e.g. `findPair({ 1.0 , 1.0 , 7.5 , 4.0, 4.1 , 3.5 },0,5)` returns 30.0 (7.5 * 4.0), which is largest product of two neighboring values.

```
public static double findPair(double[] array, int lo, int hi)
{









}
```

Write a recursive method to find the first index of the largest product of two neighboring values. e.g. `findPair({ 1.0 , 1.0 , 7.5 , 4.0, 4.1 , 3.5 },0,5)` returns 2 because 7.5x4.0=30.0 is largest product of two neighboring values.

```
public static int findPair(double[] array, int lo, int hi) {
```

6. You need to climb a flight of stairs with N steps. You can climb one or jump three steps at a time. How many different ways are there to ascend the stairs?

Process*: i) Identify the sub-problem; ii) Choose parameters and temp variables; iii) Write the base cases; iv) Write the recursive case;*

Extend your solution above to include:
1) an optional elevator 200 steps from the top.
2) a missing/broken step 15 steps from the top.
3) a non-optional worm-hole exactly 211 steps from the top.

Extend your solution above so that we only count paths that have a maximum of M moves (single steps or jumps).