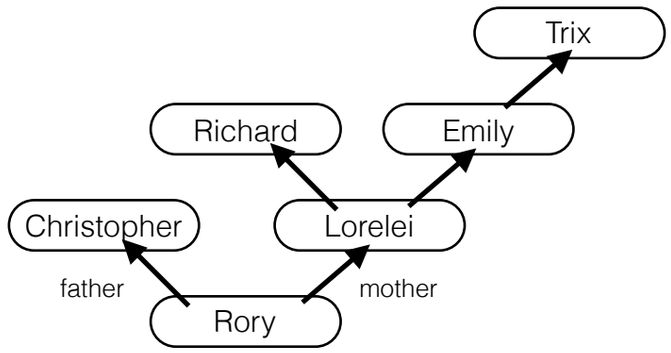
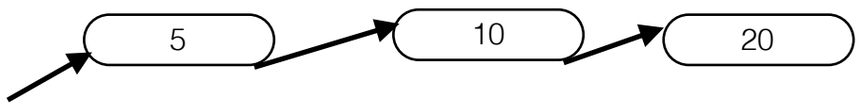
Objectives: Merge sort review; recursion: practice and more advanced;

Up next: MP7; Extra credit?



1. Write a tail recursive with a string accumulator method and a wrapper method to return the father with the longest name. Only consider the male lineage.



2. 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;
    }
}
```

3a. ***Merge Sort review:***

```
static void mergeSort(int[] data, int lo, int hi) {
   if (lo >= hi)  return;
   int mid = (lo + hi) / 2;

   mergeSort(data,?                    );
   mergeSort(data,?                    );

   int size = hi - lo + 1;
   int[] temp = new int[size];

   merge(data,temp,lo,mid+1,hi);

   for (int i = 0; i < size; i++) data[i+lo] = temp[i];
}

public static void merge(int[] a, int []tempArray,
                                  int lower, int mid, int upper){
    int tempIndex=0;
    int leftLo = lower;
    int leftHi = mid-1;
    int rightLo = mid;
    int rightHi = upper;




}
```

3b. How many levels does the merge sort activation diagram have (roughly)? Why?

3c. At each level of the tree (j =0, 1, 2, …), how many subproblems are there (as a function of N)?

3d. At each level of the tree (j =0, 1, 2, …), how many values are in the array passed into the recursive activation (as a function of N)?

4. 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;




}
```

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 FORWARD 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) {
```