22C:16 Practice Problem Set 12

Morning Section: Complete before Tuesday, 5-6-2014

Evening Section: Complete before Monday, 5-5-2014

These practice problems are all on recursion.

1. What is the output produced by the following function when it is called as
   `partition([6, 1, 9, 3, 2, 6, 1, 4], 0, 7)`

```
def partition(L, first, last):
    p = first

    for current in range(p+1, last+1):
        print L[first:p], L[p], L[p+1:current]
        if L[current] < L[p]:
            swap(L, current, p+1)
            swap(L, p, p+1)
            p = p + 1

    return p
```

2. This question is based on understanding the working of the following implementation of the quick sort algorithm.

```
def generalQuickSort(L, first, last):
    if first < last:
        p = partition(L, first, last)
        generalQuickSort(L, first, p-1)
        generalQuickSort(L, p+1, last)
```

   (a) Insert the code

```
if first > last:
    print "Base Case 0"
```

   right at the beginning of the function. How many times will we see "`Base Case 0`" printed if we make the call
   `generalQuickSort([6, 1, 9, 3, 2, 6, 1, 4], 0, 7)`

   (b) Insert the statement
   `print L[first:p], L[p], L[p+1:last+1]`
   just after the line of code `p = partition(L, first, last)` in the above function. What output do we get if make the call
   `generalQuickSort([6, 1, 9, 3, 2, 6, 1, 4], 0, 7)`

   (c) How many calls in total are made to the function `generalQuickSort` if it is called as
   `generalQuickSort([6, 1, 9, 3, 2, 6, 1, 4], 0, 7)`

   (d) Delete the second (recursive) call to `generalQuickSort` in the above definition of `generalQuickSort`. Start with a list `L = [6, 1, 9, 3, 2, 6, 1, 4]` and call this function as
   `generalQuickSort(L, 0, 7)`
   What is `L` after this call?

3. How many times is the `swap` function called (from `partition`) as a result of the call
   `generalQuickSort([6, 5, 4, 3, 2, 1], 0, 5)`

4. How many times is the `swap` function called (from `partition`) as a result of the call
   `generalQuickSort([1, 2, 3, 4, 5, 6, 7, 8], 0, 7)`

5. Write down a length-7 list sequence that causes `partition` to split the list into exactly two halves each time `partition` is called as part of the call to `generalQuickSort` on this list. In other words, the first time `partition` is called, it is called on a length-7 list, and it should split the list into two sublists of size 3 each. Subsequently, `partition` will be called on two length-3 lists. In each case, `partition` should split the list into two length-1 lists.
