Final Practice Problems (from Fall 2013, by Nick Howe)

1. Stacks and Queues

For each of the following, identify whether their behavior is stack-like, queue-like, or neither.

a.) A bag of candy

b.) Students by class year at Smith (e.g., class of ’14, class of ’15, etc.)

c.) Nodes already seen but waiting to be processed during breadth-first traversal

d.) Method calls in Java

e.) Coats of paint on a house

f.) Layers of skin

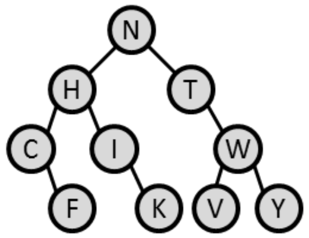
g.) A heap

h.) Cars in a parking lot

2. Trees

a.) Predict the output of the following algorithm performed on the root:

```
procedure:  tour(node)
    output data
    if right not null then tour(right) endif
    output data
    if left not null then tour(left) endif
    output data
```



b.) Suppose that the tree is a binary search tree using alphabetic order on the nodes. Where in the tree would the value M have to be inserted?

3. Recursion

Consider the following recursive procedure for determining the winner of a sporting tournament in answering the questions that follow. Note that $\lfloor x \rfloor$ indicates a *floor* operation (round down)

```
procedure:  champion($team_0$,$team_1$,…, $team_{n-1}$)
    if n == 1 then
        return $team_0$
    else
        p =  $\lfloor n/2 \rfloor$
        return winnerOfGame(champion($team_0$, …, $team_{p-1}$ ),champion($team_p$,..., $team_{n-1}$))
    endif
```

a.) In a tournament with 7 teams, how many times will the *winnerOfGame* method be called?

b.) In a tournament with 12 teams, some teams will need to win three games in order to be declared champion, while others will need to win four. Which teams (e.g., *team*$_0$, etc.) will only have to play three games?

c.) Five teams are playing a tounament: the **Holyhead Harpies**, **Wimbourne Wasps**, **Chudley Cannons**, **Tutshill Tornados**, and **Puddlemere United** (numbered in that order). Assuming that all games are won by the team whose name is first in alphabetical order, list the opponents in every game of the tournament, in the order the games would be played.

4. Sorting

Consider the method below, which implements a proposed sorting algorithm.

```java
public static void cocktail_sort(int[] arr) {
    for (int i = 1; i < arr.length/2; i++) {
        for (int j = i-1; j < arr.length-i; j++) {
            if (arr[j] > arr[j+1]) {
                int tmp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = tmp;
            }
        }
        for (int j = arr.length-i-1; j > i-1; j--) {
            if (arr[j] < arr[j-1]) {
                int tmp = arr[j];
                arr[j] = arr[j-1];
                arr[j-1] = tmp;
            }
        }
    }
}
```

a.) Determine the exact number of comparisons this algorithm would make, in terms of the number of elements in the array ***n***.

b.) How does this performance relate to other sorting algorithms we have studied?