

COP 3503 Recitation #3 Recursion Problems (Week of 6/3)
Due: 6/13/13 (Thursday) at 11:55 PM Webcourses2 time

Write complete Java programs. Please include ample comments in your code, as well as some test cases and corresponding outputs. Using Big-O notation, indicate the time complexity in terms of the appropriate variables for each of the following problems.

1) Write a recursive method to determine the number of 1s in the binary representation of a positive integer n. Here is the prototype:

```
// Precondition: n > 0.  
public static int numOnes(int n);
```

2) Write a recursive method to determine the minimum value in an integer array. (Hint: Split the array in two halves.)

```
public static int minVal(int[] numbers);
```

3) Consider a grid of nxn grid of integers like the following:

```
2 3 2 1  
1 2 2 3  
2 1 1 2  
3 1 2 3
```

In this grid, you may only move in four directions from one square to another: up, down, left and right. But, the number of the square you move from must be less than or equal to the number of the square you are moving to. Write a recursive method that determines whether or not there is a path from a starting square to an ending square. To help with this task, an auxiliary boolean array will be given as an input parameter. Each location in this array signifies whether or not a particular square has been "visited" yet. (Sorry about the poor object-oriented design. I didn't want to have to take the time to design a class with an object, etc. just for this question.)

```
// (startx, starty) is the row and column of the starting  
// location in the grid and is guaranteed to be valid.  
// (endx, endy) is the row and column of the destination  
// on the grid and is also guaranteed to be valid.  
// visited is the same size as board and stores true in  
// an element if that particular square has been reached  
// from the starting destination. The method should  
// originally be called with visited storing all falses.  
public static boolean canMove(int[][] board, int startx,  
                             int starty, int endx,  
                             int endy,  
                             boolean[][] visited);
```