In class we gave examples of iterative and recursive methods that receive n as a parameter and take time $O(f(n))$ to run, for several different functions $f(n)$. For example, the following method runs in time $O(n^2)$

```java
public static void p1(int n){
    if(n>0){
        for(int i=0;i<n;i++)
            for (int j=0;j<n;j++)
                System.out.println(i);
    }
}

public static void p2(int n){
    if(n>0){
        for(int i=0;i<n;i++)
            System.out.println(i);
        p2(n-1);
    }
}
```

For each of the following big-O running times, write an iterative and a recursive method that runs in that time (thus you will write a total of 18 methods):

a) $O(1)$  
b) $O(\log n)$  
c) $O(n)$  
d) $O(n \log n)$  
e) $O(n^2)$  
f) $O(n^2 \log n)$  
g) $O(n^3)$  
h) $O(n^3 \log n)$  
i) $O(2^n)$

Perform experiments using several different values of $n$ for every method you wrote and show the resulting running times. Do your experimental results match your analytical results?

Write a report describing your work. We are particularly interested in your observations about the behavior of different methods for each value of $n$. You may want to use graphs or plots to illustrate this. For example, the following figure illustrates the running time of a $O(n^2)$ method as a function of $n$. 

---

CS2302 Data Structures  
Fall 2013  
Lab 1  
Due Wednesday, September 11, 2013, 11:59 p.m.