Give recurrence relations to express the space complexities of these methods that are in the LinkedList class from assignments #1 and #2:

```
public void space1(
            int level, int k)
{
    rear.next= new
        ListNode(level, null);
    rear= rear.next;
    n++;
    if (k==1) return;
    space1(level+1, k/2);
    space1(level+1, k/2);
 }
```

```
public static void
      space2(
           int level, int k)
 {
    int [] A;
    if (k==1) return;
    A= new int[k];
    space2(level+1, k/2);
    space2(level+1, k/2);
 }
```

```
public static void main(String [] args)
{
        ListNode current;      LinkedList list;
        current= new ListNode(-1, null);
        list= new LinkedList(1, current, current);
        list.space1(0, 8);
        System.out.println("The list has:");
        current= list.start;
        while (current != null)
        {
            System.out.print(" " + current.data);
            current= current.next;
         }
         space2(0, 8);
  }
```

```
current= new ListNode(-1, null);
list= new LinkedList(1, current, current);
list.space1(0, 8);
```

The output is:

The list has:

-1 0 1 2 3 3 2 3 3 1 2 3 3 2 3 3