**UI：**

\*1.实现一个自动换行的ViewGroup。

\*2.简单实现网络下载图片的三级缓存。

三级缓存涉及的对象：内存，本地，网络先从内存中读取，根据内存是否存在使用，如果没有，从本地寻找对应图片加载，如果本地也没有则需要从网络中下载图片。

**算法：**

\*1.已知一个升序数组{1,2,3,4,5,6,7,8}，实现一个函数public int search(int[] num, int target)使用二分法(binary search)查找数组num中等于target的元素并返回其下标index，无相等的元素则返回-1。

\*2.已知数组{4,5,6,7,0,1,2},也可以是其他无重复元素的升序数组被旋转后的新数组例如:{5,7,8,9,1,2,3}，实现函数public int finMin(int[] num),使用二分法(binary search)查找数组num中的最小值的下标。

\*3.已知数组{4,5,6,7,0,1,2},也可以是其他无重复元素的升序数组被旋转后的新数组例如:{5,7,8,9,1,2,3}，实现函数public int search(int[] num, int target),使用二分法(binary search)查找数组num中等于target的元素并返回其下标index，无相等的元素则返回-1。

\*4.实现单向链表的翻转。

\*5.有一100阶层的楼梯，有三种走楼梯方式，一次走一阶，一次走两阶，一次走三阶。用算法实现，走完100阶总共有多少种走法。

比如:10阶的楼梯会有以下11种走法

1111111111  
111111112  
11111122  
1111222  
112222  
22222  
22231  
2221111  
2233  
22111111  
22311

**代码结构与线程同步:**

\*6.编写程序描述单车工厂、单车售卖商店。

条件:

(1)实现1个单车工厂类,1个单车售卖商店类(一个类new 两个实例),1个单车类

(2)各个商店每20秒向工厂申请5辆单车（工厂库存不足5辆就给出剩余的单车，不阻塞线程） ，商店有自己的库存(即装单车的集合)。

(3)商店随机时间（1秒到10秒）会卖出一辆单车，当商店单车库存为零时工厂会马上工厂申请5辆单车。

（提示：工厂库存不足5辆就给出剩余的单车，此时工厂如果库存为0则会阻塞线程等待工厂生产单车，商店类中有2个线程,线程1用于向工厂申请单车,线程2用于单车销售。）

(4)工厂会每2秒生产一辆单车，有自己的库存(即装单车的集合)。

补充：工厂线程先运行8秒钟后再运行商店线程，商店库存为0时停止销售单车;并向工厂申请单车直到申请成功库存>0时再开始销售

要求：实现时用好面向对象思想，代码要整洁可读性良好，结构要考虑可扩展性。