activity 的生命周期分为两部分：

一部分是典型情况下的生命周期，另一部分是异常情况下的生命后期。

前者是指在有用户参与的情况下，activity所经过的生命周期的改变

后者是指activity被系统回收或者由于当前设备的Configuration发生改变从而导致activity被销毁重建，异常情况下的生命周期的关注点和前一种情况略有不同。

典型情况下的生命周期：

正常情况下，activity会经过以下生命周期：

 （1）oncreate : 表示Activity正在被创建，这是生命周期的第一个方法。在这个方法中，做一些初始化工作，比如调用set Content View去加载界面布局资源，初始化Activity所需的数据等等。

 （2）onRestart : 表示activity正在重新启动。一般情况下，当前activity从不可见重新变为可见状态时，onRestart就会被重新调用。这种情形一般是用户行为导致的，比如用户切换到手机桌面或者用户打开一个新的activity，这时当前的activity就会暂停，也就是onPause和onStop被执行了，接着用户又回到了这个activity，就会出现这种情况。

 （3） onStart : 表示activity正在被启动，即将开始，这时activity已经可见了，但是可还没有出现在前台，还无法与用户进行交互。这个时候其实可以理解为activity已经显示出来了，但是我们还看不到。

 （4） onResume : 表示activity已经可见了，并且出现在前台并开始活动了。要注意这个和onStart的对比，onStart和onResume都表示activity已经可见，但是onStart的时候activity还在后台，onResume的时候activity才显示前台。

   (5) onPause : 表示activity正在停止，正常情况下，紧接着onStop就会被调用。在特殊情况下，如果这个时候快速地再回到当前activity，那么onResume就会被调用。笔者的理解是，这种情况是属于极端情况，用户操作很难重现这一场景。此时可以做一些存储数据，停止动画等工作，但是不能太耗时，因为这会影响影响到新activity的显示，onPause必须先执行完，新activity的onResume才会执行。

（6） onStop：表示activity即将停止，可以做一些稍微重量级的回收工作，同样不能太耗时。

（7） onDestory：表示Activity即将被销毁，这是Activity生命周期中的最后一个回调，在这里，我们可以做一些回收工作和最终的资源释放。

=================================================

(1 )activity第一次启动，回调如下：onCreate---》onStart---》onResume()

(2 )当用户打开新的activity时，回调如下：onPause---》onStop，如果新activity是透明的，则不会回调onstop

(3 )回到原activity时，会回调：onRestart---》onStart--->onResume

(4 )当用户按back键回退时，回调如下：onPause--》onStop--》onDestory

封闭 单一 最少知道 依赖 替换 接口分离

/////////////////////////启动流程---------------------

Activity：startActivity方法的真正实现在Activity中  
Instrumentation：用来辅助Activity完成启动Activity的过程  
ActivityThread

（包含ApplicationThread + ApplicationThreadNative + IApplicationThread）：真正启动Activity的实现都在这里

调用startActivty实际上最后还是调用了startActivityForResult 方法

1：真正执行activity的启动过程的是ActivityManagerNative.getDefault().startActivity。  
2：执行启动activity之后，有一个代码检查结果。  
我们先看看第2个方法：  
Instrumentation#checkStartActivityResult

启动Activity实际上也是进行了进程间的通信

通过我们的intent数据，结合classLoader找到了相关的类并且加载进来了

不是系统回收而是虚拟机去回收，每次我们运行一个app的时候，Android系统就会给我们的虚拟机（Dalvik/art）分配一定大小的内存，随着app运行可以申请更大内存（不同手机各不相同，有的手机能给到200多M）；最后来回答您的问题，回收顺序为空进程-->后台Activity-->服务进程-->可见Activity -->前台Activity。也就是说如果你的A界面不可见了，转为后台Activity，回收的会更快点；而如果A界面还是可见，而回收的可能稍慢点。