第一题：

1. 正整数数列全部由3组成
2. 将一个数分成两部分和，对两部分分别递归，遍历所有的分割方法，将最大的作为本次递归的返回值，可以通过记忆化搜索的方式进行优化
3. 见1.py

第二题：

指数增长很快

第三题：

['人羊狼菜', '狼菜', '人狼菜', '狼', '人羊狼', '菜', '人羊菜', '羊', '人羊', '成功']

['人羊狼菜', '狼菜', '人狼菜', '狼', '人羊狼', '羊', '人羊', '成功']

['人羊狼菜', '狼菜', '人狼菜', '菜', '人羊狼', '羊', '人羊', '成功']

['人羊狼菜', '狼菜', '人狼菜', '菜', '人羊菜', '羊', '人羊', '成功']

以上是所有的不走回头路的答案

代码见3.py

第四题：

见4.py

第五题：

感觉对于精度的影响不是很大，因为精度是由循环中的eps控制的，但是迭代次数还收受到了初始值的影响，答案和初始值越接近，那么迭代次数就会较少

第六题：

同上。感觉除的多一点会更加接近答案，让迭代次数更少

第七题：

res=res-(res\*res\*res-x)/(3\*(res\*res))

见7.py

第八题：

第一种方法是蒙特卡洛。收敛极其缓慢

第二种方法是arctanx的泰勒展开，收敛速度比上面一种好很多

第三种方法是卡马努金给出来的式子，好快的收敛，据说有线性复杂度

第九题：

见9.py