第一章：

问题：

Fibo:兔子两个月后每个月生一只=>当月出生=两个月前

当月总数=当月出生+上月数量

递归 f(n)=f(n-1)+f(n-2) 复杂度：O（2^n/2）

T(n) = T(n-1) + T(n-2) + **O(1)复杂度要加一个O（1）,因为递归调用本身需要计算**

复杂度计算方法：递推

动态规划 for(i=2;i<=n;++i){fu=f1+f2;f1=f2;f2=fu} 复杂度O(n)

概念：计算复杂度

Case: f(n) = f(n-1) + f(n-2) + ... f(1) + 1

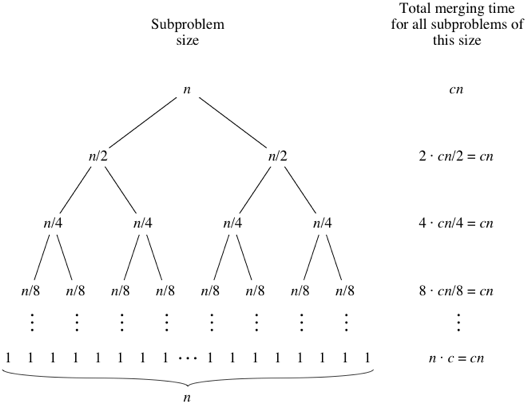
So: f(n-1) = f(n-2) + ... f(1) + 1

错位相减: f(n) = f(n-1) + f(n-1) = 2 f(n-1)=>2^n

归并排序复杂度：<https://softwareengineering.stackexchange.com/questions/297160/why-is-mergesort-olog-n>

F(n){f(2/n);f(2/n) ;merge(xxx)}

T(n)=T(n/2)+T(n/2)+O(n)**+1** ;1这个常量的复杂度确实存在的但是在计算O（n）是可以省略的



1这个常量复杂度在叶子节点的复杂度是 n/2往上是n/4

叶子节点的数量是 2^(log(2)n)=n

第二章：

线性表：顺序表和链表

问题：链表合并，链表倒置

优缺点：链表增删快，顺序表随机存储快

栈：

队列：

第三章

树：二叉多叉，平衡非平衡

概念：度，平衡，满

问题：先/中/后序遍历，简单递归

中序非递归：栈

层次非递归：队列

计算树深度：递归

计算树节点总数：递归

二叉排序树：左大又小，不平衡，avl通过旋转普通2叉树实现

哈夫曼树：编码，选出最小的两个节点…

堆排序：pass

第四章

图

概念：度，完全图

遍历：深度广度，栈，堆

最小生成树，prim,Kruskal

路径：dijkstra

第六章

查找，折半，**顺序，hash，分块**

Hash碰撞:链表，再hash

排序：

Core：快排，堆排