1. 应用分治法的三个基本步骤
①分解问题(divide)：把原问题分解为若干个与原问题性质相类似的子问题
②求解子问题(conquer)：不断分解子问题直到可方便求出子问题的解为止
③合并子问题的解(combine)：合并子问题的解得到原问题的解
归并排序(Merge Sort) O(nlgn)
快速排序算法分析
最坏情况：
∵ 二个无序区的大小分别为n-1和0
∴ T(n) = θ(n2)
最好情况：
T(n)= O(nlgn)
2.应用分治法解题的三个基本步骤为：
①divide：把具有n个元素的数组分解为二个n/2大小的子数组
②conquer：递归地分解子数组，直到子数组只包含一个元素为止
③combine：二二合并已排好序的子数组使之成为一个新的排好序的子数组，重复这样二二合并的过程直到得到原问题的解
3. 分治法适用条件
①原问题可以分解为若干个与原问题性质相类似的子问题
②问题的规模缩小到一定程度后可方便求出解
③子问题的解可以合并得到原问题的解
④分解出的各个子问题应相互独立，即不包含重叠子问题
如求Fib数问题
4. Master方法
条件：如果T(n)具有如下递归形式：
T(n)=aT(n/b)+f(n)，其中a≥1，b>1且为常数，f(n)为渐进函数
定理4.1 Master定理
对于非负递归函数T(n)=aT(n/b)+f(n)，其中a ≥1，b>1且为常数，f(n)为渐进函数，则T(n)有如下三种渐进界限：

5.红黑树首先是一棵二叉查找树，所以二叉查找树的所有性质红黑树都具有，此外还有自身五个特性：
①每个结点要么是黑色要么是红色
②树根结点的颜色为黑色
③叶结点(nil)为黑色
④如果某个结点为红色，则它的左、右孩子结点均为黑色
⑤对树中任一结点，所有从该结点出发到其叶结点的路径中均包含相同数目的黑色结点
外部结点(external node):不包含实际关键字
内部结点(internal node):包含实际关键字
RB-delete算法时间：
∵ 红黑树的高度h=O(lgn)
Tree-delete算法时间为O(lgn)
RB-delete-fixup算法时间为O(lgn)
∴ RB-delete算法时间为O(lgn)
RB-Insert算法分析：
∵ 红黑树的高度h=O(lgn)

Tree_Insert算法时间为：O(lgn)
RB-Insert-fixup算法时间为：O(lgn)
∴RB-Insert算法时间为：O(lgn)
6.数据结构扩张的步骤
1. 挑选一个合适的基本数据结构
2. 决定在基本数据结构上应增加的信息
3. 修改基本数据结构上的操作并维持原有的性能
4. 修改或设计新的操作
区间树和序统计数理解书上的定义和操作即可
7. 动态规划方法的基本步骤
1. 描述问题的最优解(optimal solution)结构特征
2. 递归定义最优解值
3. 自底向上计算最优解值
4. 从已计算得到的最优解值信息中构造最优解
8. 动态规划的要素
1. 最优子结构性质
最优子结构性质是应用动态规划方法的必要前提，即所解问题必须具有最优子结构性质才能用动态规划方法求解。
所谓最优子结构性质是指一个问题的最优解中所包含的所有子问题的解都是最优的。
对于一个问题发现其最优子结构性质的几个因素：
①检查问题的解所面临的选择
②假定某种选择为一最优解
③分析这种选择将产生多少子问题
④证明子问题的解也是最优的
最优子结构的细节问题：
特别需要指出的是当问题本身不具有最优子结构性质时不能滥用最优子结构性质。
2. 重叠子问题

虽然问题分解后是否存在重叠子问题不是应用动态规划方法的必要前提，但存在重叠子问题应用动态规划方法可以提高算法效率
9.动态规划相关算法
装配线调度问题(Assembly-line Scheduling)：Fastest-Way算法的时间为Θ(n)
矩阵链乘(Matrix-chain Multiplication)：算法matrix-chain-order包含三重循环
又∵每重循环的次数≤n ∴算法时间为$O(n^3)$。
最长公共子序列(Longest Common Subsequece)：算法LCS-length(X,Y)见书P210
算法时间为Θ(mn)
最优二叉查找树(Optimal Binary Search Tree)：算法时间为$O(n^3)$

10 贪心法的基本思想
每次选择总是选出当前最优的解
Recursive-Activity-Selector(s,f,i,j)：算法总时间为O(nlgn)
应用贪心法的几项工作
①确定问题的最优子结构性质
②将优化问题转化为作出一种选择，即贪心选择
③贪心选择后应只留下一个子问题，其它子问题均为空
④证明贪心选择的正确性，即本次贪心选择与剩余子问题的最优解可以构成原问题的最优解
贪心选择性质是应用贪心法求解的一个必要条件。所
谓选择性质是指所求问题的最优解可以通过一系列局部
最优的贪心选择而得到。即：局部最优=>全局最优
最优子结构性质也是应用贪心法求解的一个必要条件。

局部最优+一系列贪心选择产生原问题的一个最优解。

11. 胚（matroids)
def：一个胚应是满足下述条件的有序对M=(S,I)
①S是有穷非空集
②I是S的一个非空独立子集族，使得若B∈I且
A ⊆ B，则A∈I。满足此性质的I具有遗传性，即B独立，B的子集亦独立，其中Φ∈I。
③若A∈I，B∈I且|A|<|B|，则存在一个元素x∈B-A，使得A∪{x} ∈I，则称M具有交换性。
最大独立子集
给定一个胚M=(S,I)，对于I中一个独立子集A∈I，若S中有一个元素x不属于A，使得将x加入A后仍保持A的独立性，即A∪{x}∈I，则称x为A的一个可扩张元素（extension）。
当胚中的一个独立子集A没有可扩张元素时，称A是一个最大独立子集。
最优子集
使w(A)权值达到最大的独立子集A称为最优子集。
例：求连通网络G=(V,E)的最小生成树问题。
定义加权胚$M_G=(S_G,I_G)$的权为w'为：
$w'(e)=W_0-w(e)>0$，其中e∈E，W_0为大于G中任一边权值的常数
令A是G的生成树，则$w'(A)=(|V|-1)W_0-w(A)$
若w'(A)最大则w(A)最小，所以A是G的MST
胚的应用即任务调度问题看书理解

12平摊分析方法
平摊分析是指在某种数据结构上完成一系列操作，在最坏情况下所需的平均时间。
只需要掌握势能法
势能法(potential method)：
思想：通过定义一个势函数完成对每个操作的平摊核定
令n个操作的数据结构为D，数据结构的初态为D_0
C_i：表示第i次操作的实际成本
$C_i^\wedge$：表示第i次操作的平摊成本
D_i：表示在数据结构状态为D_{i-1}上完成第i次操作后的数据结构状态，op_i
即： $D_{i-1}====>D_i$ i=1,2,…,n
Φ：势函数
Φ(D_i)：表示将Di映射为一实数，这个实数值称为势能
用势函数Φ表示第i次操作的核定费用$C_i^\wedge$定义如下：
$C_i^\wedge= C_i+Φ(D_i)-Φ(D_{i-1})= C_i+Φ_i-Φ_{i-1}$
其中$Φ_i-Φ_{i-1}$称为势差，势差的不同结果类似与记帐法：
$Φ_i-Φ_{i-1}>0$：超额收费
$Φ_i-Φ_{i-1}<0$：收费不足
如同记帐法需对操作的平摊核定进行验证，为保证势函数的正确性，也需验证平摊总费用是否是实际总费用的上界，即：

13. 动态表(Dynamic tables)
表扩张(table expansion)：发生在插入一个元素x时，此时表满没有空间存放x，所以将引起表扩张。表扩张的步骤为：

①申请一块更大表
②拷贝原表到新表中
③释放原表
④在新表中插入x
表收缩（table contraction）：发生在删除一个元素x后，表中元素个数小于某个设定值时，引起表收缩。表收缩步骤为：
①删除元素x
②申请一块更小的表
③拷贝原表到新表中
④释放原表
定义一个装填因子(load factor)α(T)描述动态表当前的状态：
①非空表：α(T)=num[T]/size[T]　　0≤α≤1
num[T]：表中元素个数，size[T]：表的大小
②空表：size[T]=0，此时定义α(φ)=1
注意空表指的是表的大小为0的情况

具体例题可参考17-4.3

14. 二项树(Bin0mial trees)
1. 二项树定义
def：仅包含一个结点的有序树是一棵二项树称为B_0树。二项树B_k由二棵B_{k-1}树组成，其中一棵B_{k-1}树的根作为另一棵B_{k-1}树根的最左孩子($k\geq 0$)。
2. 二项树性质
引理19.1　二项树B_k具有性质：
①有2^k个结点　　$n=2^k$
②树的高度为k　　$k=\lg n$
③深度为i处恰好有个结点　　$0\leq i\leq k$
④根的度最大且为k，若根的孩子从左到右编号为k-1,k-2,…,1,0，则孩子i恰好是子树B_i的根。
二项堆(binomial Heaps)
1. def：二项堆H是一个满足下述条件的二项树的集合：
①H中的每棵二项树满足最小堆性质
②对任意的非负整数k，H中至多有一棵二项树根的度为k
2. 二项树的逻辑表示
例：H中有13个结点，即n=13，则$B_H=1101_2$
head[H]------->B_0-------->B_2-------->B_3
二项堆中的所有二项树由一根表链接，根表的头指针用head[H]表示。
①根表为一单链表
②根表链接所有二项树的根结点
③根表按照度的递增序链接

过程	二叉堆(最坏情况)	二项堆(最坏情况)	斐波那契堆(平摊)
MAKE-HEAP	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$
INSERT	$\Theta(\lg n)$	$\Omega(\lg n)$	$\Theta(1)$
MINIMUM	$\Theta(1)$	$\Omega(\lg n)$	$\Theta(1)$
EXTRACT-MIN	$\Theta(\lg n)$	$\Theta(\lg n)$	$O(\lg n)$
UNION	$\Theta(n)$	$\Omega(\lg n)$	$\Theta(1)$
DECREASE-KEY	$\Theta(\lg n)$	$\Theta(\lg n)$	$\Theta(1)$
DELETE	$\Theta(\lg n)$	$\Theta(\lg n)$	$O(\lg n)$

15. Fib堆

1. def Fib堆是一具有最小堆有序的树的集合。

Fib堆和二项堆比较:

相似点:

①均支持5个基本操作和2个扩展操作
②均为树的集合
③堆中每棵树均满足最小堆性质
④均采用根表链接堆中所有树根结点

不同点:

①Fib堆根表中度不再唯一
②Fib堆中树不要求一定是二项树
③Fib堆根表不需要按照度的递增序排列
④Fib堆根表头指针指向根表中具有最小值的树根结点
⑤Fib堆操作的时间分析采用平摊分析方法

所有Fib堆操作均采用统一的势函数定义如下:

$\Phi(H)=t(H)+2m[H]$

其中: t(H)表示堆中树的个数
m[H]表示mark域标记为true的结点数

16. 最大流(Maximum Flow)

Ford-fulkerson方法

Ford-Fulkerson-Method(G,s,t)
初始化流f=0
while 存在增广路径P
do 沿增广路径P增加流f
return f

算法时间为: $O(E|f^*|)$.

Edmonds-Karp算法求最大流的时间为$O(VE^2)$。
二分图最大匹配算法时间为O(VE')=O(VE)

17其他需要看书理解的重点内容

1. 贪心选择性质的证明
2. 动态规划中矩阵链乘法，最长子序列
3. 胚的应用，即带惩罚的活动选择
4. 序统计数，区间树的性质和相关操作
5. Ford-fulkerson方法求最大流

祝大家考试成功，鸡年大吉吧!