

数据结构与算法 总复习

数据结构与算法

3555

5大算法策略

分治法
贪心法
动态规划
回溯法
分支限界法

3大数据结构

线性表（栈、队列）
树（二叉树、森林）
图（广义表）

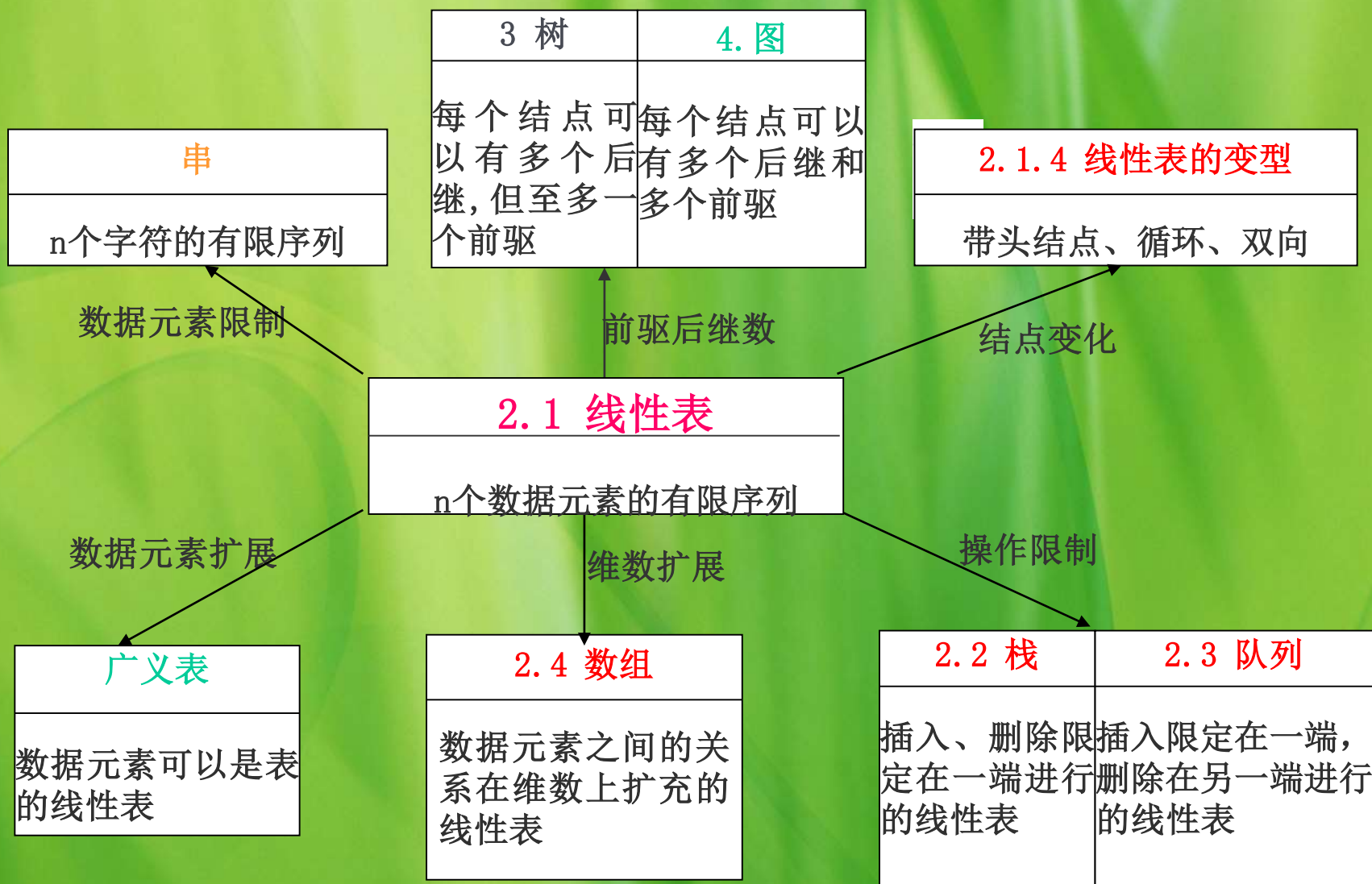
常用操作

遍历

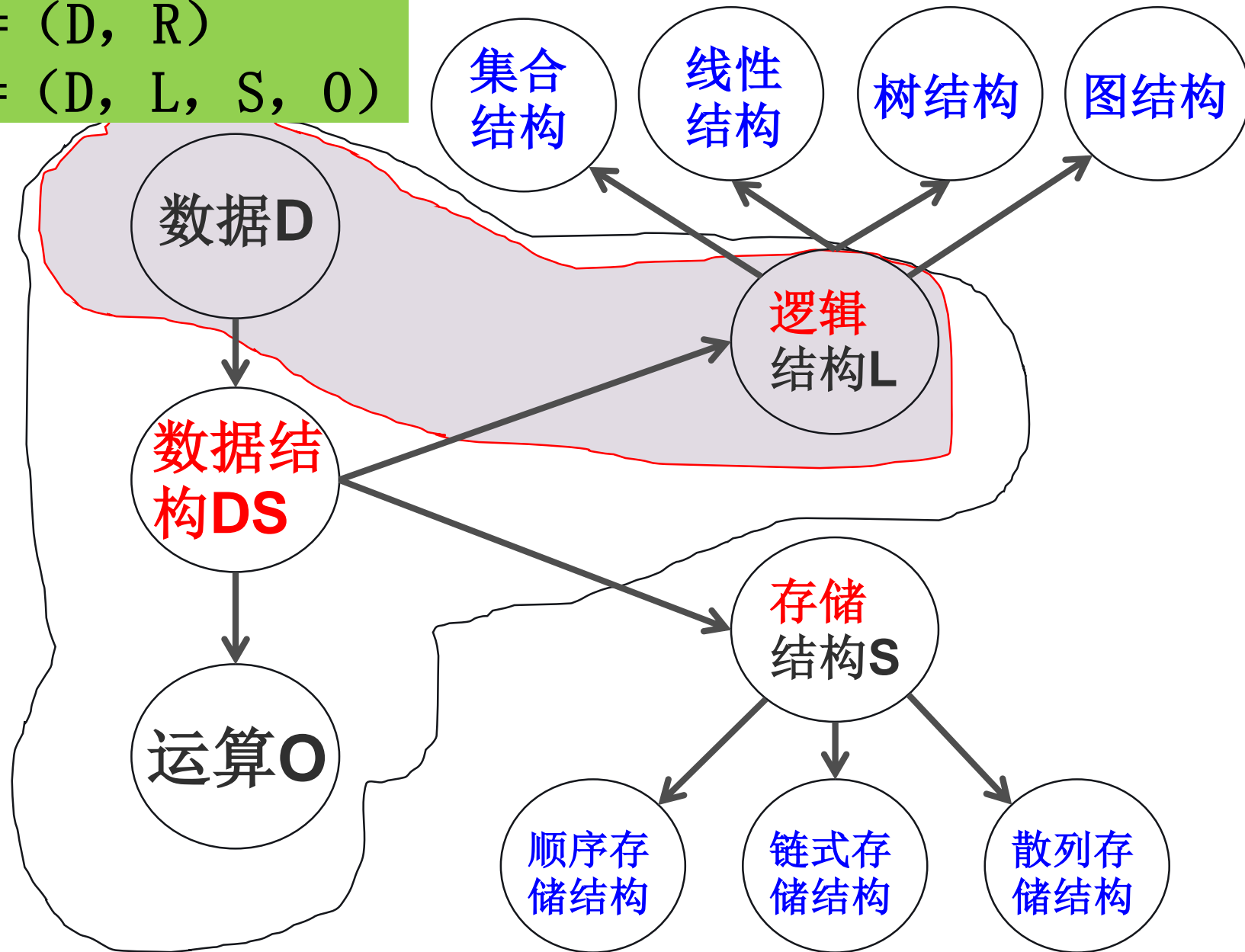
5大查找方法

5大排序方法

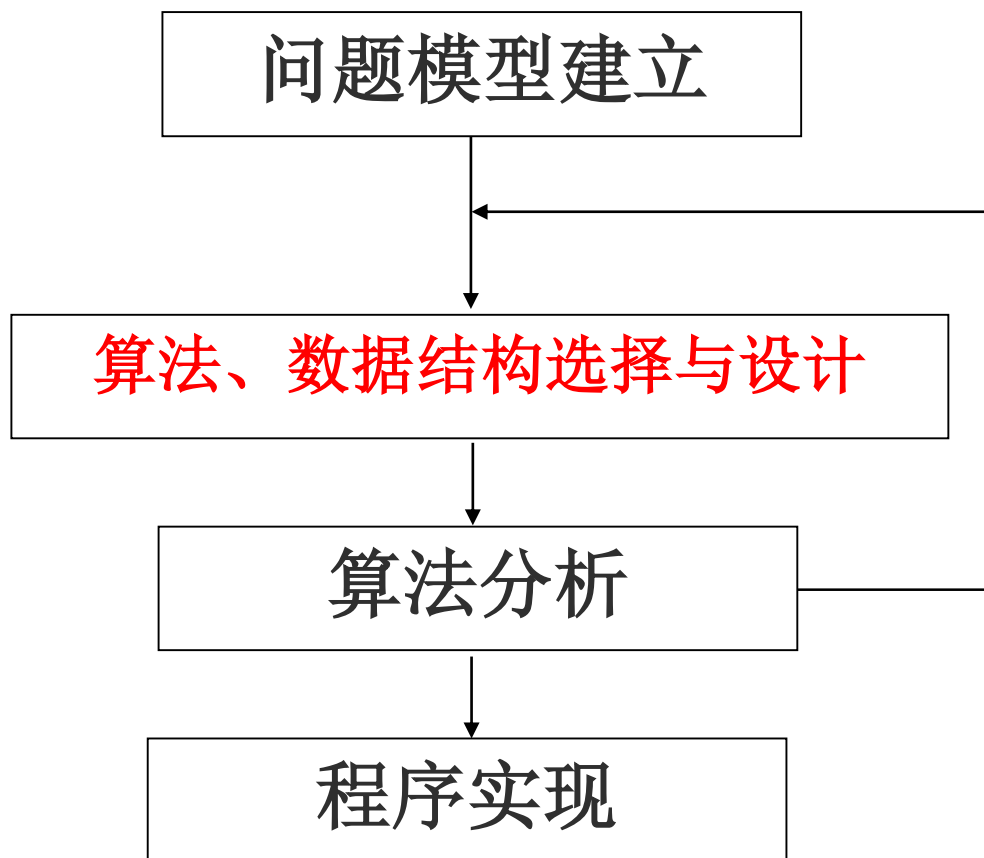
数据结构关联图



$DS = (D, R)$
 $DS = (D, L, S, O)$



计算机问题求解过程



问题模型建立

K最大问题：统计最热门的10个搜索词

$K \ll n$ 时：

K与n相当时：

最短工期问题：关键路径（概念） \rightarrow 关键活动：活动的最早开始时间=活动的最晚开始时间

装载问题：容量为 M 吨的轮船， n 个集装箱的重量分别为 $\{w_1, w_2, \dots, w_n\}$ ，相应的价值分别为 $\{v_1, v_2, \dots, v_n\}$ ，每种集装箱最多能装1箱。

算法、数据结构选择与设计

算法： 算法是输入转换为输出的一系列步骤

或者算法是求解特定问题的步骤的有限序列。

算法的4个特性： 有穷性、确定性、可行性、功能性（输入、输出）

算法与数据结构的关系：

算法分析：

算法与程序的区别：

算法

1. 算法设计：数据结构、解决问题切入点（本质、原因、不让问题出现）、策略、步骤；

2. 算法实现：

C语言函数形式描述：函数名、参数表、返回值；

函数架构：初值部分、主体部分、尾部处理；

递归算法：初值部分在调用函数中赋值，由参数表传入，递归出口部分，处理部分、递归调用部分；

3. 算法分析：正确性、时间复杂度、空间复杂度。

术语（一）

完全二叉树

排序二叉树（二叉排序树**BST**）

平衡二叉树（**AVL**）

判定二叉树（二分查找）

哈夫曼树（最优二叉树、带权路径长度最短，哈夫曼编码）

术语（二）

Prim（普里姆最小生成树算法）

Kruskal（克鲁斯卡尔最小生成树算法）

AOV网（顶点表示活动的网，回路，拓扑排序）

AOE网（边表示活动的网，最短工期，关键路径）

Dijkstra（迪杰斯特拉单源点最短路径算法：按路径长度递增次序产生源点到各顶点的最短路径）

Floyed（弗洛伊德所有顶点对最短路径算法）

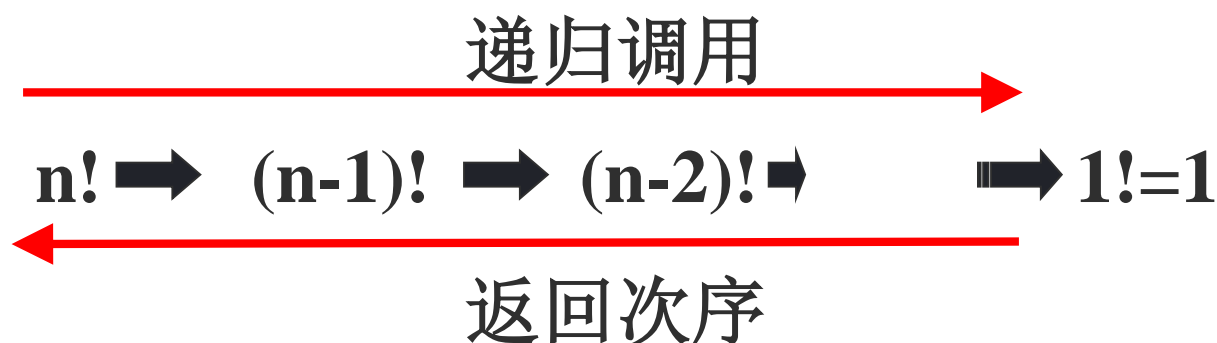
Shell（希尔排序算法）

术语（三）

遍历（**DFS**、**BFS**）、遍历的具体化；
递归

递归过程与递归工作栈

- 递归过程在实现时，需要自己调用自己。
- 层层向下递归，返回次序正好相反：



递归函数的内部执行过程

- (1) 运行开始时，首先为递归调用建立一个工作栈，其结构包括值参、局部变量和返回地址；
- (2) 每次执行递归调用之前，把递归函数的值参和局部变量的当前值以及调用后的返回地址压栈；
- (3) 每次递归调用结束后，将栈顶元素出栈，使相应的值参和局部变量恢复为调用前的值，然后转向返回地址指定的位置继续执行。

1. 概念、术语、性质；
2. 基本运算；
3. 存储结构；
4. 基本运算实现举例；
5. 应用举例。

顺序与链式存储结构比较

顺序存储结构的特点

- 逻辑上相邻的元素，其物理位置也相邻；
- 可随机存取表中任一元素，是**随机存取**存储结构；
- 按最大可能长度预分存储空间，是一种**静态**存储结构；
- 插入删除时，需移动大量元素，等概率下：插入平均移动元素为 $n/2$ ；删除平均移动元素为 $(n-1)/2$

链式存储结构的特点

- 逻辑上相邻的元素，其物理位置不一定相邻；元素之间的邻接关系由指针域指示；
- **是非随机存取**存储结构；对链表的存取必须从头指针开始
- 是一种**动态**存储结构；
- 插入删除运算非常方便；只需修改相应指针值

链式存储结构：

结点结构：每个结点的各个域：名字、作用；

特殊的结点：头结点、层结点（广义表多元多项式）；

整体结构：头指针、总的头指针；

链式存储与顺序存储结合：如：图的邻接表；

子问题解可合并特性
无重叠子问题

分治法

深度优先搜索
+回溯

回溯法

贪心选择特性
最优子结构特性

贪心法

5大算
法策略

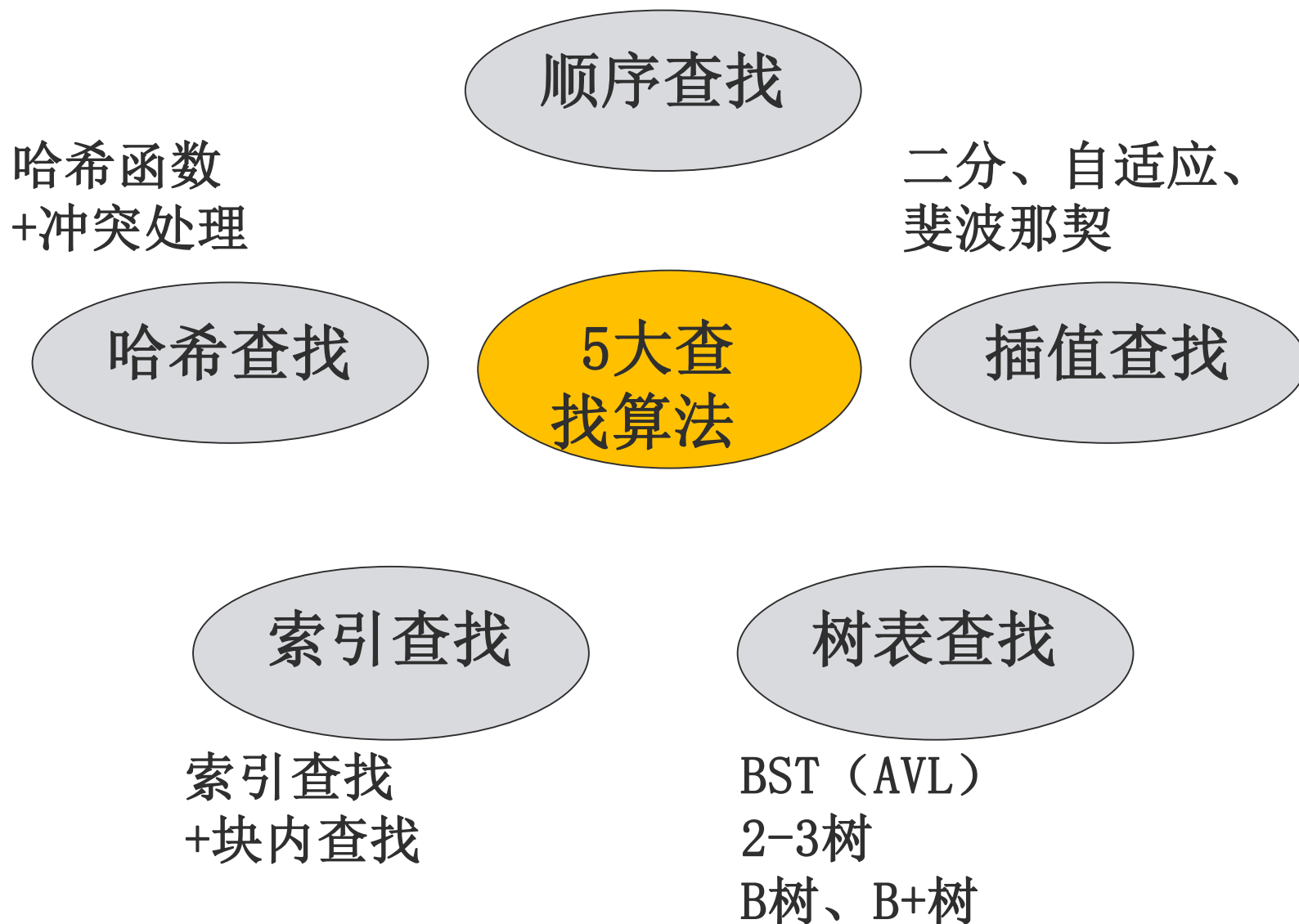
分支
限界法

广度优先搜索
+剪枝策略

动态
规划法

最优子结构特性
无后效性
有重叠子问题

查找方法、查找表、平均查找长度ASL



方法、时间复杂度、
稳定性

三步：定位、腾位、到位；
直接、折半、**希尔**

插入排序

两步：分配、收集；
链式基数

基数排序

两步：比较、交换；
冒泡、改进冒泡、**快速**

5大排
序算法

交换排序

归并排序

2个或多个有序子序列
归并为一个有序序列

选择排序

两步：选择、加入；
简单、树形、**堆**

各种排序算法的比较

排序算法	平均时间复杂度	最坏情况下 时间复杂度	空间复杂度	稳定性
直接插入排序	$O(n^2)$	$O(n^2)$	$O(1)$	稳定
希尔排序	猜测 $O(n^{7/6})$? $O(n^2)$ -- $O(n^{4/3})$	$O(1)$	不稳定
冒泡排序	$O(n^2)$	$O(n^2)$	$O(1)$	稳定
快速排序	$O(n\log_2 n)$	$O(n^2)$	$O(\log_2 n)$	不稳定
简单选择排序	$O(n^2)$	$O(n^2)$	$O(1)$	稳定
归并排序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n)$	稳定
堆排序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(1)$	不稳定
基数排序	$O(n)$	$O(n)$	$O(RADIX)$	稳定

动态规划求解0-1背包问题

给定 n 种物品和一背包。物品 i 的重量是 $w_i > 0$ ，其价值为 $v_i > 0$

背包的容量为 C 。如何选择装入背包的物品，使得装入背包中物品的总价值最大且总重量小于 C ？

装入方法：每种物品只有两种选择，即装入或不装入，不能将物品多次装入，也不能只装入一部分。

变形问题：

一辆存储容量为 C 的卡车。 N 种水果单箱的重量分别为 w_i ，每种水果单箱的价值分别为 v_i ，每种水果最多能装载1箱，问如何装箱才能获得最多收益？

一货船载重量为 C ，现有 N 种货物要装船运输，每种货物的重量分别为 w_i ，每种货物的价值分别为 v_i ，每种货物该船最多能装载1件。如何装载货物获得最大价值？

有物品重量与价值2个维度，目标是选择满足重量小于C的最大价值的问题。

1. 分析：定义 $m(i, j)$ = 背包容量为 j ，由 $1, \dots, i$ 个物品装填背包问题的最优值。边界条件 $i=0$ 时， $m(i, j) = 0$;

$w_i > j$ 时，舍去物品 i ， $m(i, j) = m(i-1, j)$ 。

其他：

情形 1： $m(i, j)$ 不选择第 i 个物品： $m(i, j) = m(i-1, j)$ ，即在重量限制为 j 下， $\{1, \dots, i-1\}$ 个物品装填背包所产生的最大价值；

情形 2： $m(i, j)$ 选择第 i 个物品：新的重量限制为 $= j - w_i$

$m(i, j) = m(i-1, j - w_i) + V_i$ ，即 $m(i-1, j - w_i)$ 为新重量限制下， $\{1, \dots, i-1\}$ 个物品装填背包所产生的最大价值。

此时有： $m(i, j) = \max$ （情形1，情形2）

2. 定义递归关系式：

$$m(i, j) = \begin{cases} 0 & \text{if } i = 0 \\ m(i-1, j) & \text{if } w_i > j \\ \max\{m(i-1, j), V_i + m(i-1, j - w_i)\} & \text{otherwise} \end{cases}$$

3. 动态规划列表求解： 例子

		j + 1 →											
		0	1	2	3	4	5	6	7	8	9	10	11
n + 1 ↓	∅	0	0	0	0	0	0	0	0	0	0	0	0
	{ 1 }	0	1	1	1	1	1	1	1	1	1	1	1
	{ 1, 2 }	0	1	6	7	7	7	7	7	7	7	7	7
	{ 1, 2, 3 }	0	1	6	7	7	18	19	24	25	25	25	25
	{ 1, 2, 3, 4 }	0	1	6	7	7	18	22	24	28	29	29	40
	{ 1, 2, 3, 4, 5 }	0	1	6	7	7	18	22	28	29	34	34	40

$$m(i, j) = \begin{cases} 0 & \text{if } i = 0 \\ m(i-1, j) & \text{if } w_i > j \\ \max \{m(i-1, j), v_i + m(i-1, j - w_i)\} & \text{otherwise} \end{cases}$$

物品	价值	重量
1	1	1
2	6	2
3	18	5
4	22	6
5	28	7

4. 答：最优解：{ 4, 3 }
最优值 = 22 + 18 = 40

C = 11

考试题型

2014年A卷

- 一、选择题（共**20**分，共 **10**题，每题**2** 分）
- 二、填空题（共**12**分，共 **8**题，每空**1** 分）
- 三、简答题（共**14**分）
- 四、应用题（共**30**分）
- 五、算法与程序题（共**24**分）

1. 有关二叉树下列说法正确的是 ()

- A. 二叉树的度为2 B. 一棵二叉树的度可以小于2
C. 二叉树中至少有一个结点的度为2 D. 二叉树任何一个结点的度都为2

2. 树的后根遍历序列等同于该树对应的二叉树的 ()

- A. 先序序列 B. 中序序列 C. 后序序列 D. 层次遍历序列

3. 具有 n 个顶点的无向图至少应有 () 条边才可能是一个连通图

- A. n B. $n-1$ C. $n+1$ D. $n-2$

4. 下列哪一种图的邻接矩阵是对称矩阵

- A. 有向图 B. 无向图 C. AOV网 D. AOE网

5. 当在一个有序的顺序存储表上查找一个数据时，既可用折半查找，也可用顺序查找，但前者比后者的查找速度()

- A. 必定快 B. 必定慢 C. 在大部分情况下要快
D. 取决于表递增还是递减

6. 下面说法错误的是()

- (1) 算法时间复杂度的含义是指额外的辅助空间
(2) 在相同的规模 n 下，算法的执行次数为 n 的算法在时间上总是优于执行次数为 $2n$ 的算法
(3) 所谓大 O 时间复杂度是指最坏情况下，估算算法执行时间的一个上界
(4) 同一个算法，实现语言的级别越高，执行效率就越低

- A. (1) B. (1), (2) C. (1), (4) D. (3)

7. 一个算法应该是()

- A. 程序 B. 问题求解步骤的描述
C. 要满足五个基本特性 D. A和C.

8. 线性表采用单链表存储结构。单链表中结点分成2部分：数据元素与指针。其中指针的准确含义是（ ）。

- A. 内存地址 B. 数组下标 C. 下一元素在内存中的地址 D. 左、右孩子地址

9. 在双向循环链表中, 在p指针所指向的结点前插入一个指针q所指向的新结点, 其修改指针的操作是（ ）。

注: 双向链表的结点结构为 `struct node {int data; struct node * llink, * rlink}`。

- A. `p->llink=q; q->rlink=p; p->llink->rlink=q; q->llink=q`
B. `p->llink=q; p->llink->rlink=q; q->rlink=p; q->llink=p->llink;`
C. `q->rlink=p; q->llink=p->llink; p->llink->rlink=q; p->llink=q;`
D. `q->llink=p->llink; q->rlink=p; p->llink=q; p->llink=q;`

10. 对于一个具有n个顶点e条边的无向图, 若采用邻接表表示, 则表头向量的大小为（ ）。

- A. n B. n+1 C. n-1 D. n+e

二、填空题（共12分，共 8题，每空1 分）

1. 在下面的程序段中，对 x 的赋值语句的最坏时间复杂度（ ）

```
for(int i=0;i<n;i++)  
    for(int j=0;j<i;j++)  
        for(int k=0;k<j;k++)  
            x=x+delta;
```

2. 顺序存储结构是通过（ ）表示元素之间的关系的;链式存储结构是通过（ ）表示元素之间的关系的。

3. 在等概率的情况下，在有 n 个元素的顺序表中插入一个结点需平均移动（ ）个结点,删除一个结点需平均移动（ ）结点。

4. 二叉树的叶子结点个数 n_0 与度为2的结点个数 n_2 的关系是（ ）

二、填空题（共12分，共 8题，每空1 分）

5. 完全二叉树适合采用()存储结构,查找第*i*个元素的左孩子的公式为()

6. 已知二叉树的前序遍历序列和中序遍历序列分别是 **ABDGCEFH**和**DGBAECFH**,它的后序遍历序列是 ()

7. 对于含*N*个顶点*E*条边的无向连通图，利用**Prim**算法生成最小代价生成树其时间复杂度为()，利用**Kruskal**算法生成最小代价生成树其时间复杂度为()。

8. 假定有*k*个关键字互为同义词，若用线性探测再散列法把这*k*个关键字存入散列表中，至少要进行 () 次探测。

三、简答题（共14分）

1、简述动态规划与分治递归的相同与不同点。(6分)

答：动态规划算法与分治法类似，其基本思想也是将待求解问题分解成若干个子问题，先求解子问题，然后从这些子问题的解得到原问题的解。（2分）

与分治法不同的是，适合于用动态规划求解的问题，经分解得到子问题往往不是互相独立的。（2分）。

若用分治法来解这类问题，则分解得到的子问题数目太多，有些子问题被重复计算了很多次。如果我们能够保存已解决的子问题的答案，而在需要时再找出已求得的答案，这样就可以避免大量的重复计算，节省时间。（2分）

三、简答题（共14分）

2、如果在一个顺序表中对关键字序列采用顺序查找法和二分查找法查找一个数据，该关键字序列分别应该满足什么要求？
(4分)

答：用顺序查找法：关键字可以按任意的次序排列，有序无序均可实现。（2分）

采用二分查找法：关键字序列必须按从小到大或从大到小的顺序排列，而且只能顺序存储。（要求有序 1 分，顺序存储 1 分）

三、简答题（共14分）

3、设输入序列为2，3，4，5，6，利用一个栈能得到序列2，5，3，4，6吗？栈可以用单链表实现吗？(4分)

答：不能得到序列2，5，3，4，6（2分）。栈可以用单链表实现，这就是链栈。由于栈只在栈顶操作，所以链栈通常不设头结点。（2分）

四、应用题（共30分）

1、给定数组 $a[1...n]$ ， n 为偶数。设计一个算法，在最坏情况下用不超过 $3n/2 - 1$ 次比较找出 $a[1...n]$ 中元素的最大值和最小值。（6分）

答：将 $a[1]...a[n]$ 两两分成一组，即为 $a[1]$ 和 $a[2]$ ， $a[3]$ 和 $a[4]$ ， $a[5]$ 和 $a[6]$ ，... 共计 $n/2$ 组，（1分）

每组中的两个元素进行比较，得到最大值和最小值，共计 $n/2$ 次比较；（2分）

上述每组中的最大值组成数组 $b[1...n/2]$ ，冒泡排序(或其他排序方法)求得 $b[1...n/2]$ 中的最大值，即为原数组 $a[1...n]$ 的最大值，需 $n/2 - 1$ 次比较；（1分）

同理，上述每组中的最小值组成数组 $c[1...n/2]$ ，冒泡排序(或其他排序方法)求得 $c[1...n/2]$ 中的最小值，即为原数组 $a[1...n]$ 的最小值，需 $n/2 - 1$ 次比较；（1分）

因此，共需 $3n/2 - 2$ 次比较（1分）

四、应用题（共30分）

2、求解下列递归关系式。(6分)

(1) $T(n)=3T(n/2) + 5n$ (3分)

(2) $f(n)= 3f(n/5) + 6n$ (3分)

答： 求解递归关系式方法没讲。

四、应用题（共30分）

3、请对报文“**data structure**”中的每一个字符进行二进制编码（空格不编码），使得保证报文正确译码的情况下编码最短，并给出相应的最优二叉树。（6分）

答：（1）统计各个字符出现次数如下：

d:1 s:1 c:1 e:1 a:2 r:2 u:2 t:3 (1分)

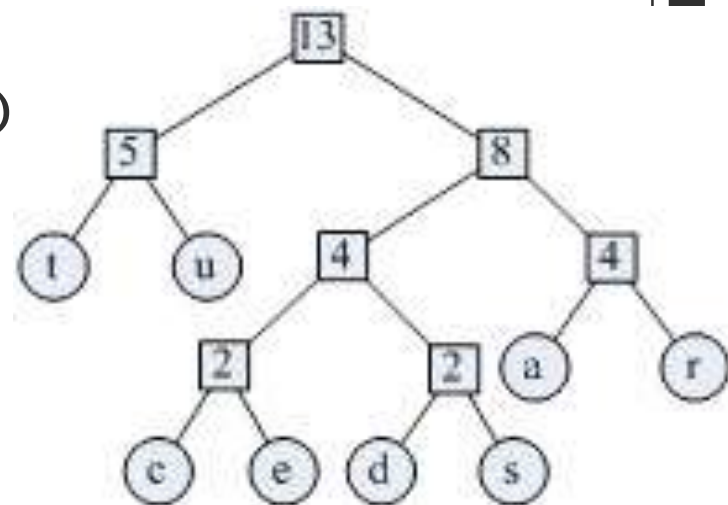
（2）画出最优二叉树（哈夫曼树，不唯一）（3分）

（3）根据左0右1对哈夫曼树的编码如下：

t:00 u:01 c: 1000 e: 1001

d: 1010 s: 1011 a: 110 r: 111 (2分)

（编码和哈夫曼树要对应）



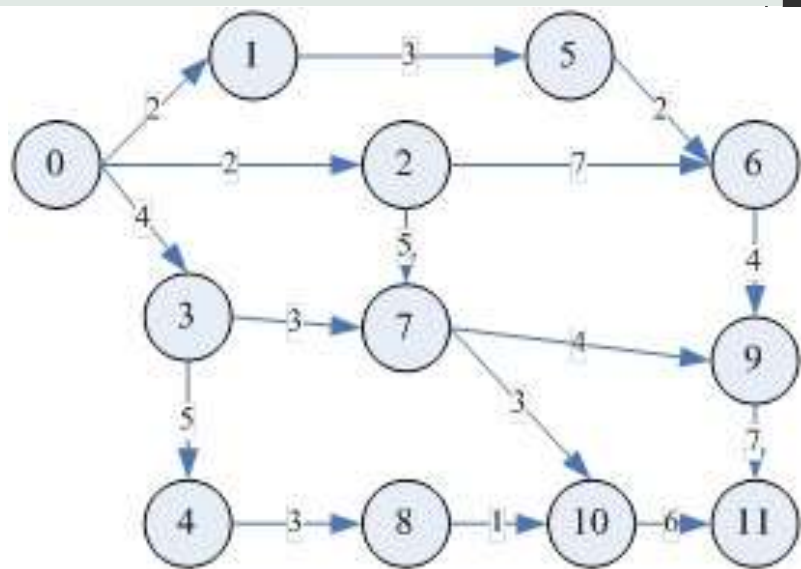
哈夫曼树及编码：

给出一组报文，要求对其进行编码，使得保证报文能正确译码，且整个编码长度最短（最优二叉树）。

方法：

1. 统计列出各字符在报文中出现的频度；
2. 按哈夫曼方法画出哈夫曼树；
3. 对各字符按左分支0，右分支1，进行编码；
4. 从根到叶，给出各字符的相应编码；
5. 计算出带权路径长度。

4、某一工程作业的网络图如图所示，其中箭头表示作业，箭头上的数字表示完成作业所需的天数。箭头前后的圆圈表示事件，圆圈中的数字表示事件的编号。用事件编号的序列（例如**0-3-4-8-10-11**）表示进行作业的路径。求（1）完成此工程的关键路径；（2）完成此工程所需的最少天数（6分）



答：（1）**0-2-6-9-11** （4分）错1条路径扣1分
（2）**20天** （2分）

关键路径（最短工期问题）：

给出一个图，求关键路径（关键活动）、最短工期。

方法：关键路径（概念）→ 关键活动（活动的最早开始时间 e =活动的最晚开始时间 l ）。

不要直接给出关键路径或最短工期，要有求解过程：

计算结果为：

顶点	ev	lv	活动	弧	持续T	e	l	l-e	关键活动
V1	0	0	a1	<1,2>	3	0	1	1	
V2	3	4	a2	<1,3>	2	0	0	0	a2
V3	2	2	a3	<2,4>	2	3	4	1	
V4	6	6	a4	<2,5>	3	3	4	1	
V5	6	7	a5	<3,4>	4	2	2	0	a5
V6	8	8	a6	<3,6>	3	2	5	3	
			a7	<4,6>	2	6	6	0	a7
			a8	<5,6>	1	6	7	1	
	Max +	Min -							

四、应用题（共30分）

5、现有一文件F含有**1000**个记录，其中只有少量记录次序不对，且它们距离正确位置不远；如果以比较和移动次数作为度量，那末将其排序最好采用什么方法？为什么？（6分）

答：采用直接插入排序算法（3分），因为记录序列已基本有序，直接插入排序比较次数少，且由于少量次序不对的记录与正确位置不远，使直接插入排序记录移动次数也相对较少，故选直接插入排序算法（3分）。

算法与程序题（共24分）

1、某软件公司今年计划完成 n 个软件项目中的部分项目，项目 j 在 s_j 月开始 f_j 月结束，项目 j 具有 v_j 效益。若两个项目不重叠，即时间不交叉，则这两个项目是可以安排执行的，即兼容的。现给定项目序列集合 $A = \{a_1, a_2, a_3, a_4\}$ ，对应的开始时间集合 = {3月, 1月, 6月, 5月}，结束时间集合 = {6月, 5月, 8月, 12月}，效益集合 = {5, 3, 7, 5}（万元）在确定时间1年的12个月内，找到一项目子集计划执行，使得所执行的该项目子集产生最大效益值且每个项目互相兼容。

（共12分）

（1）用动态规划方法求解该问题，需要写出算法思想步骤和递归关系式。（8分）

（2）求解该问题的最优解和最优值。（4分）

答：待解。

动态规划求解相容活动安排问题：

活动集合 $A = \{a_1, a_2, a_3, a_4\}$,

开始时间集合 $S = \{3月, 1月, 6月, 5月\}$,

结束时间集合 $F = \{6月, 5月, 8月, 12月\}$,

效益集合 $V = \{5, 3, 7, 5\}$

使得：最大效益值且互相兼容。

分析、递归关系式，最优解和最优值

答：定义 $p(j)$ = 最大的与项目 j 兼容的项目序号 i , $i < j$

定义 $OPT(j)$ = j 个项目 $1, 2, \dots, j$ 所获得的最大价值即最优解 (1分)

情况1: OPT 中选择第 j 个项目:

不能采用不兼容的项目 $\{p(j)+1, p(j)+2, \dots, j-1\}$; 必须包括剩余的 $1, 2, \dots, p(j)$ 个项目构成的最优解。 (2分)

情况2: OPT 不选择第 j 个项目:

必定包括项目 $1, 2, \dots, j-1$ 所形成的最优解 (2分)

$$OPT(j) = \begin{cases} 0, & j = 0 \\ \max\{v_j + OPT(p(j)), OPT(j-1)\}, & other \end{cases}$$

动态规划求解相容活动安排问题:

活动集合 $A = \{a_1, a_2, a_3, a_4\}$,

开始时间集合 $S = \{3月, 1月, 6月, 5月\}$,

结束时间集合 $F = \{6月, 5月, 8月, 12月\}$,

效益集合 $V = \{5, 3, 7, 5\}$

使得: 最大效益值且互相兼容。

分析、递归关系式, 最优解和最优值

j	1	2	3	4
P(j)	0	0	1	2
OPT (j)	5	3	12	8

最优解为 a_1 和 a_3 , 最优值为12.

算法与程序题：2、如果允许在循环队列的两端都可以进行插入和删除操作。要求：（1）写出循环队列的类型定义；
（2）写出“从队尾删除”和“从队头插入”的算法。（共12分）

答：[设计] 用一维数组 $v[0..M-1]$ 实现循环队列，其中 M 是队列长度。设队头指针 $front$ 指向队头元素的前一位置，队尾指针 $rear$ 指向队尾元素。定义 $front=rear$ 时为队空， $(rear+1)\%m=front$ 为队满。约定队头端入队向下标小的方向发展，队尾端入队向下标大的方向发展。（2分）

（1）**#define M** 队列可能达到的最大长度
typedef struct
 { **elemtp data[M];**
 int front, rear;
 } **cycqueue;**（2分）

算法与程序题2、如果允许在循环队列的两端都可以进行插入和删除操作。要求：（1）写出循环队列的类型定义；
（2）写出“从队尾删除”和“从队头插入”的算法。（共12分）

答：（2）**elemtp delqueue (cycqueue Q)**

//Q是如上定义的循环队列，本算法实现从队尾删除，若删除成功，返回被删除元素，否则给出出错信息。

{ if (Q.front==Q.rear) {printf(“队列空”); exit(0);}

Q.rear=(Q.rear-1+M)%M; //修改队尾指针。

return(Q.data[(Q.rear+1+M)%M]); //返回出队元素。

}//从队尾删除算法结束

（4分）错1个语句扣1分

算法与程序题2、如果允许在循环队列的两端都可以进行插入和删除操作。要求：（1）写出循环队列的类型定义；
（2）写出“从队尾删除”和“从队头插入”的算法。（共12分）

答：（2） `void enqueue (cycqueue Q, elemtp x)`
// Q是顺序存储的循环队列，实现“从队头插入”元素x。
`{if (Q.rear==(Q.front-1+M)%M) {printf(“队满”；exit(0);}`
 `Q.data[Q.front]=x; //x 入队列`
 `Q.front=(Q.front-1+M)%M; //修改队头指针。`
`}// 结束从队头插入算法。`
（4分）错1个语句扣1分

考试题型

2016年A卷

一、选择题（共20分，每小题2分）

二、填空题（共10分，每小题2分）

三、简答题（共14分）

三、应用题（5题，共40分），30

四、算法与程序题（每小题10分，共30分），24

一、选择题（每小题 2分，共20 分）

1. 与邻接矩阵相比，图的邻接表存储方式更适合（ ）。

A. 有向图 B. 稠密图 C. 有权图 D. 稀疏图

2. 二叉排序树的（ ）结果是一个关键字的递增有序序列

A. 先序遍历 B. 中序遍历 C. 后序遍历 D. 层次遍历

3. 下列哪个算法没有应用贪心策略（ ）

A. AOV网络拓扑排序算法 B. dijkstra算法 C. prim算法 D. huffman编码算法

4. 下列哪个排序算法没有应用分治策略（ ）

A. 快速排序 B. 冒泡排序 C. 归并排序 D. 希尔排序

5. 在循环双向链表中，在p结点之后插入s结点的操作是（ ）。

A. $p \rightarrow next = s$; $s \rightarrow pre = p$; $p \rightarrow next \rightarrow pre = s$; $s \rightarrow next = p \rightarrow next$;

B. $p \rightarrow next = s$; $p \rightarrow next \rightarrow pre = s$; $s \rightarrow pre = p$; $s \rightarrow next = p \rightarrow next$;

C. $s \rightarrow pre = p$; $s \rightarrow next = p \rightarrow next$; $p \rightarrow next = s$; $p \rightarrow next \rightarrow pre = s$;

D. $s \rightarrow pre = p$; $s \rightarrow next = p \rightarrow next$; $p \rightarrow next \rightarrow pre = s$; $p \rightarrow next = s$;

6. 图的广度优先搜索算法是借助（ ）这种线性结构上实现的。

A. 顺序表 B. 单链表 C. 栈 D. 队列

7. 二叉树的先序序列是EFHIGJK，中序序列为HFIEJKG，该二叉树右子树的根是（ ）。

A. E B. F C. G D. H

8. 一棵完全二叉树有1001个结点，则其叶结点的个数是（ ）。

A. 500 B. 501 C. 502 D. 以上答案都不对

9. 对线性表进行二分查找时，要求线性表必须（ ）

A. 以顺序方式存储

B. 以链接方式存储

C. 以顺序方式存储，且结点按关键字有序排序

D. 以链接方式存储，且结点按关键字有序排序

10. 数据序列（2，1，4，9，8，10，6，20）只能是（ ）的两趟排序后的结果。

A. 快速排序

B. 冒泡排序

C. 选择排序

D. 插入排序

二、填空题（每小题2分，共10分）

1. 已知输入数据个数为 n ，下面程序段中的时间复杂度是()
 $i=n*n$; $\text{while}(i!=1)\{ i/= 2;\}$
2. ABCDE五个数据依次进栈,输出BDECA,假设X表示入栈操作,Y表示出栈操作,则得到该输出的操作序列为 ()
3. 已知链队列的头尾指针分别是f和r, 则将值x入队的操作序列是()。
4. 动态规划算法与分治算法最大的区别就是, 适合用动态规划算法求解的问题通常具有 () 性质。
5. 分别采用堆排序, 快速排序(第一个元素为枢纽), 冒泡排序和归并排序, 对初态为有序的表, 则最省时间的是()算法, 最费时间的是()算法。

$O(\log n)$, XXYXXYXYYY

$p=\text{new node}(x)$; $r\text{->next}=p$; $r=p$;

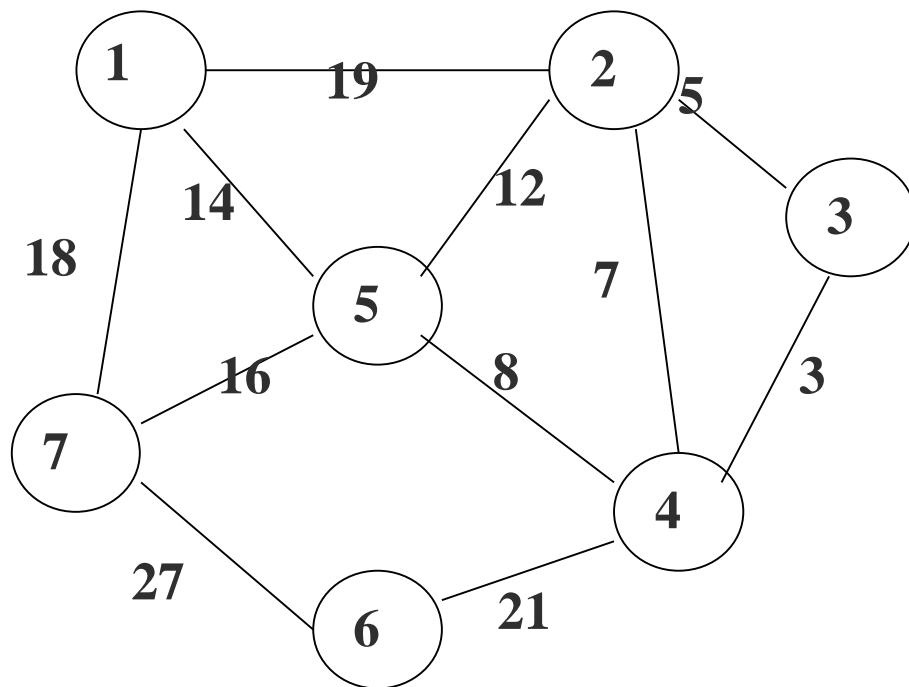
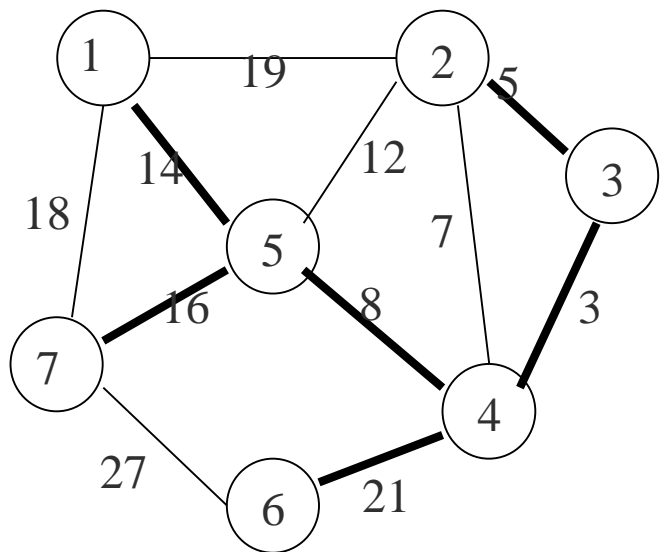
重叠子问题; 冒泡排序、快速排序

三、应用题（共5题，共40分）

1、应用**Prim**算法求出下图中从结点1开始的最小生成树，要求画出最小生成树并给出求解中添加至最小生成树的边的先后顺序。**(5分)**

答：添加边的先后顺序为：

(1, 5), (5, 4), (4, 3), (3, 2), (5, 7),
(4, 6) (3分)



2、将关键字序列为(7,8,30,11,18,9,14)存储到哈希表中，哈希表的存储空间是一个下标从0开始的一维数组，哈希函数是 $H(\text{key})=(\text{Key} \times 3) \text{MOD } 7$ ，处理冲突采用线性探测再散列法，要求装填因子为0.7。请画出所构造的哈希表，并计算等概率情况下查找成功的平均查找长度。（10分）

答：关键字序列有7个数据，装填因子0.7，因此哈希表的长度是 $7/0.7=10$ (2分)

存储地址	0	1	2	3	4	5	6	7	8	9
关键字	7	14		8		11	30	18	9	
冲突次数	1	2		1		1	1	3	3	

（每个关键字正确存放0.5分，共3.5分；每个关键字的冲突次数0.5分，共3.5分）

$ASL(\text{成功}) = (4+2+3 \times 2) / 7 = 12/7$ （1分）

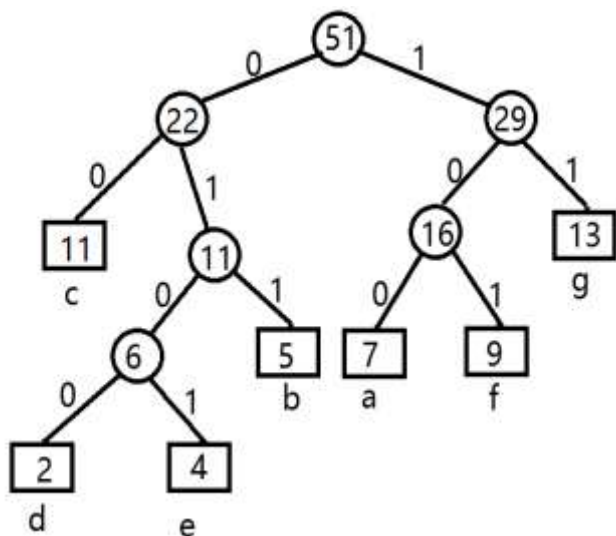
3. 用主方法求解递归式 (1) $T(n)=4T(n/3)+n^2$,
(2) $T(n)=4T(n/3)+n$,
(3) $T(n)=9T(n/3)+n^2$ 的渐近增长率 (10分)

答：没讲，不考。

4、某文本中a,b,c,d,e,f,g七个字符出现的次数分别为7,5,11,2,4,9,13, 根据huffman树为每个字符编码, 要求画出huffman树, 给出每个字符的编码, 并计算编码后的文件长度 (10分)

答: 文件编码长度 = $2*(11+13)+3*(7+5+9)+4*(2+4) = 48 + 63 + 24 = 135$ (2分)

(huffman树正确4.5分; 每个字符编码正确0.5分, 共3.5分)



字母编号	对应编码	出现频率
a	100	7
b	011	5
c	00	11
d	0100	2
e	0101	4
f	101	9
g	11	13

5、将关键字序列{12, 2, 16, 30, 28, 10, 16*, 20, 6, 18}建成大顶堆, 给出该大顶堆序列。(5分)

答: (30,28,16,20,18,10,16*,2,6,12)
(每个关键字在正确位置0.5分, 共5分。)

堆问题: 给一个关键字序列, 大顶堆还是小顶堆? 给出堆序列(画出堆图)、给出输出堆顶或输出几次堆顶后的堆图(堆序列)。

方法:

1. 按关键字序列建完全二叉树;
2. 从完全二叉树的最后一个非叶结点开始筛选建初始堆;
3. 输出堆顶, 再进行筛选, 将余下元素调整成新堆;

四、算法与编程题（每小题10分，共30分）

1. 设计算法删除非递减的有序单链表中重复的元素。例如：（7， 10， 10， 21， 30， 42， 42， 42， 51， 70）将变作（7， 10， 21， 30， 42， 51， 70），并分析算法的时间复杂度。（10分）

在非递减的有序单链表中，删除数值相同的元素，需知道被删除元素结点的前驱结点。算法需要遍历整个单链表，如果链表结点个数为 n ，则复杂度为 $O(n)$ 。(1分)。

LinkedList DelSame (LinkedList la) { (1分)

||a是非递减的有序单链表，算法去掉数值相同的元素，使表中不再有重复的元素。设la是带头结点的单链表。

```
if (la->next) { // 带头结点的单链表非空
```

```
pre=la->next;  p=pre->next;
```

||p是工作指针，**pre**是p的同步前驱结点的指针。（初始化2分）

while (p!=null) (1分)

```
if (p->data==pre->data)    //处理相同元素值的结点
```

{u=p; pre->next=p->next; p=p->next;free (u) ; } (3分) ||释放相同元素值的结点

else { pre=p; p=p->next; } (2分) //处理前驱, 后继元素值不同

} } DelSame

四、算法与编程题（每小题10分，共30分）

写出二叉树的二叉链表存储结构的结构体定义，设计算法并编程计算二叉树的高度。（10分）

```
typedef struct BiNode （1分）
    TElemType data; （1分）
    struct BiNode *lchild,*rchild; （2分）
}BiNode,*BiTree; （1分）

int Height(BiTree T) （1分）
{
    if(T==NULL) return 0; （2分）
    else return 1 + max{Height(T->lchild), Height(T->rchild)}; （2分）
}
```

遍历是一种抽象操作，遍历的具体化可以实现许多功能。如计算树的高度、叶结点、单分枝、双分枝结点个数、数据大于某个值的结点数（最大、最小值）。

3、 某公司**2017**年有 **n** 个候选工程项目，项目 **i** 将在 **s_i** 月份开始并在 **f_i** 月份结束，预计可为公司带来收益 **v_i** ， **$1 \leq i \leq n$** 。但是该公司人员有限，同一时间只能开建一个项目，若两个项目 **i** 和 **j** 建设周期不重叠，即 **$f_i \leq s_j$** 或 **$f_j \leq s_i$** ，则该公司可同时选中这两个项目进行建设。问该公司选中哪些项目进行建设能够获得最大预期收益。

(a) 用动态规划方法求解该问题，给出分析过程和递归关系式。

(b) 给定**8**个候选项目的开始月份分别为**(2,4,1,5,4,6,7,9)**，结束月份分别为**(5,6,7,8,9,10,11,12)**，预期收益分别为**(12,20,23,13,26,20,11,16)**，请根据递归式求解这个具体问题，要求给出求解过程表格、最优解和最优值。

$S=(2,4,1,5,4,6,7,9),$

$F=(5,6,7,8,9,10,11,12),$

$V=(12,20,23,13,26,20,11,16),$

递归式、求解过程表格、最优解和最优值。

定义 $p(j)$ =最大的与项目 j 兼容的活动序号 $i, i < j$ (1分)

定义 $OPT(j)$ = j 个项目 $1,2,...,j$ 所获得的最大权重，即最优解

情况1: OPT 中选择第 j 个项目

不能选择与项目 j 不兼容的项目 $\{p(j)+1, p(j)+2, ..., j-1\}$ ，必须包括剩余的 $1, 2, ..., p(j)$ 个项目构成的最优解。 (1分)

情况2: OPT 不选择第 j 个项目

必定包括项目 $1, 2, ..., j-1$ 所形成的最优解 (1分)

递归表达式为:

$$OPT(j) = \begin{cases} 0, & j = 0 \\ \max\{v_j + OPT(p(j)), OPT(j-1)\}, & other \end{cases}$$

$S=(2,4,1,5,4,6,7,9),$
 $F=(5,6,7,8,9,10,11,12),$
 $V=(12,20,23,13,26,20,11,16),$

j	1	2	3	4	5	6	7	8
P(j)	0	0	0	1	0	2	3	5
OPT (j)	12	20	23	25	26	40	34	42

答：选中项目5和项目8，能够获得最大预期收益42