

# 2024 年上海海事大学攻读硕士学位研究生入学考试试题

(重要提示: 答案必须做在答题纸上, 做在试题上不给分)

考试科目: 828 考试科目名称: 计算机综合

## 一、选择题 (每题 2 分, 共 30 分)

1. 10 阶对称矩阵顺序压缩存储至一维数组  $A[n]$  中, 则数组  $A$  长度至少为 ( )。

- A.100                      B.40                      C.55                      D.80

2. 5000 个待排序关键字, 需要在最快时间内找到最小的 10 个关键字, 则采用 ( )。

- A.快速排序              B.堆排序              C.归并排序              D.插入排序

3. 树  $T$  中边的集合为  $\{(A,B), (A,C), (A,D), (B,E), (C,F), (C,G)\}$ , 将该树转化为二叉树, 其深度为 ( )。

- A.3                      B.4                      C.5                      D.6

4.  $m$  个顶点的无向图, 至少需有 ( ) 条边才能保证该图是连通的。

- A. $m-1$               B. $(m-1)(m-2)/2+1$               C. $m(m-1)/2$               D. $m^2$

5. 若经常按序号查找线性表中的元素, 则采用 ( )。

- A.顺序存储              B.链式存储              C.静态存储              D.静态存储或链式存储

6. 若要在  $O(n \log n)$  的时间内完成排序, 且要求稳定, 则选择 ( )。

- A.快速排序              B.堆排序              C.归并排序              D.直接插入排序

7. 栈的输入序列为  $\{1, 2, 3, 4, 5, 6\}$ , 则输出序列为 ( )。

- A.5 3 4 6 1 2              B.3 2 5 6 4 1              C.3 1 2 5 4 6              D.1 5 4 6 2 3

8. 串的长度为  $n$ , 模式串的长度为  $m$ , 则执行简单模式匹配算法的最坏时间复杂度为 ( )。

- A. $O(n*m)$               B. $O(n)$               C. $O(n^2)$               D. $O(n+m)$

9. 对于  $n$  个待排序关键字, 则快速排序需要 ( ) 个辅助记录单元。

- A.1                      B. $n$                       C. $n \log n$                       D. $\log n$

10. 在一个无向图的邻接表中, 其中边表结点有  $m$  个, 则图中边的个数为 ( )。

- A. $m$                       B. $m/2$                       C. $2m$                       D. $m^2$

11. 并发进程失去封闭性是指 ( )。

- A.多个相对独立的进程以各自的速度向前推进  
B.并发进程的执行结果与速度无关  
C.并发进程执行时，在不同时刻发生的错误  
D.并发进程共享系统的资源
- 12.死锁的四个条件中不能被破坏的是（ ）。
- A.互斥条件      B.不可剥夺条件      C.请求并保持条件      D.循环等待条件
- 13.SPOOLing 系统可以实现设备的（ ）分配。
- A.虚拟      B.共享      C.独占      D.物理
- 14.下列关于快表的说法，不正确的是（ ）。
- A.利用空间换时间技术  
B.快表存储在主存中  
C.快表是高速缓存存储器  
D.快表中存储的是部分页表
- 15.银行家算法是死锁的（ ）阶段。
- A.死锁预防      B.死锁避免      C.死锁检测      D.死锁解除

## 二、填空题（每空 2 分，共 30 分）

- 1.评价一个好的算法应从\_\_\_\_\_，可读性，健壮性，高效性，存储空间少这些方面考虑。
- 2.设循环队列 Q 采用数组 Elem(n)为存储空间，front 为队头，rear 为队尾，则队列中元素个数为\_\_\_\_\_。
- 3.一组初始记录关键字序列为{45, 80, 40, 22, 48, 78}，按照递增进行排序，则直接插入排序的第一趟序列为\_\_\_\_\_。
- 4.用一维数组存放一棵二叉树如下表所示，则中序遍历序列为\_\_\_\_\_。
- |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | b | c | ^ | d | ^ | e | ^ | ^ | f | g | ^ | ^ | h | i | ^ | ^ | ^ | ^ | j |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
- 5.已知一颗二叉树的中序序列为 CBEDFAGH，后序序列为 CEFDBHGA，则先序序列为\_\_\_\_\_。
- 6.广义表((a),((b),c),(((d))))的长度为\_\_\_\_\_。

7. 下列程序段的时间复杂度为\_\_\_\_\_。

```
i=1;
```

```
while(i<=n)
```

```
    i=i*5;
```

8. 给定权值分别为 3,9,6,2,5 的结点构造哈夫曼树，则哈夫曼树的带权路径长度为\_\_\_\_\_。

9. 若  $a=6$ ,  $b=5$ ,  $c=3$ ,  $d=2$ ,  $e=8$ , 则后缀表达式  $abc-/de*+=$ \_\_\_\_\_。

10. 哈夫曼树叶结点数为  $n$ , 用二叉链表存储结构, 则哈夫曼树共\_\_\_\_\_个空指针域。

11. 计算机系统中有 7 个并发进程共享系统资源, 每个进程需要资源最大为 3 个, 为了确保不死锁, 则最少需要\_\_\_\_\_个资源。

12. 若有  $N$  个进程共享同一临界资源, 如果利用信号量机制来实现这些进程对资源的互斥访问, 信号量值的变化范围为\_\_\_\_\_。

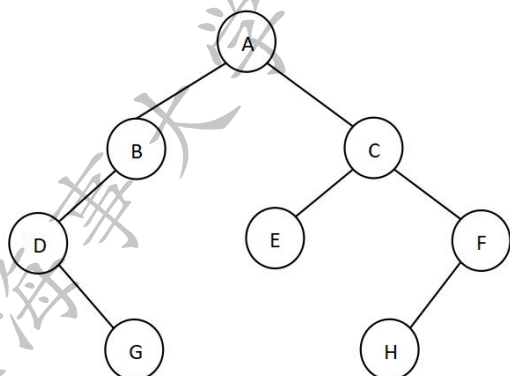
13. 若要使作业平均周转时间最短, 则应用\_\_\_\_\_调度算法。

14. 执行一次磁盘读写操作所花费的时间包括\_\_\_\_\_、旋转延迟时间、传输时间。

15. OS 的接口包括命令接口, 图形界面接口和\_\_\_\_\_。

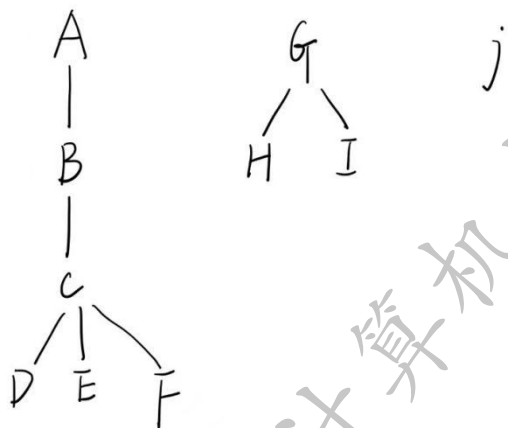
### 三、应用题 (每题 5 分, 共 60 分)

1. 已知二叉树如下图所示, 画出该树的后序线索二叉树



2. 设散列表为  $HT[0..12]$ ，散列函数为： $H(key) = 2 * key \% 13$ ，采用链地址法解决冲突，若插入的关键码序列为  $\{1, 13, 51, 46, 22, 53, 67, 65, 34\}$ ，试画出该散列表。

3. 将下图所示的森林转换为二叉树。



4.关键字序列{32, 6, 50, 27, 97, 15, 92, 29, 20}, 按关键字值递增的次序进行排序, 试回答下列问题

(1) 以第一个元素为基准的快速排序第一趟扫描后的结果

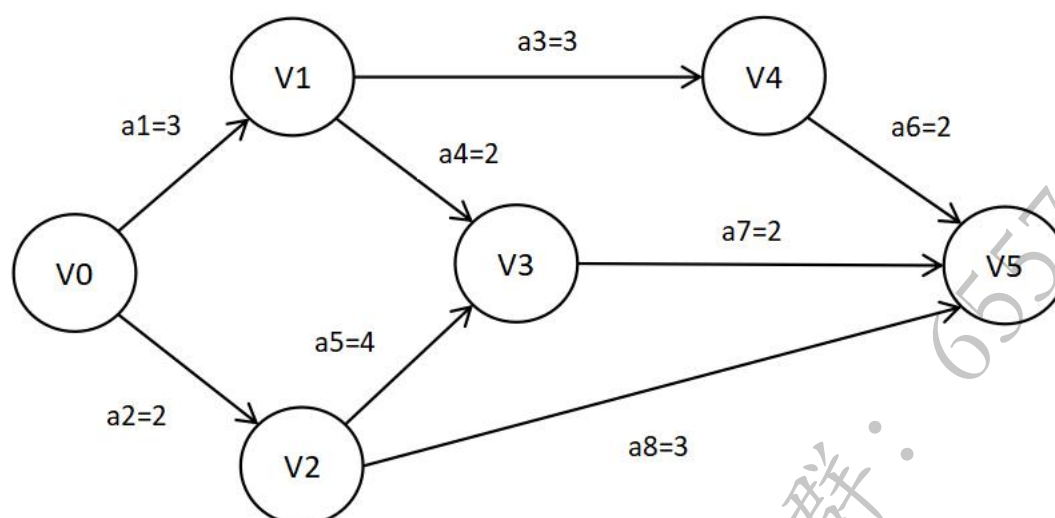
(2) 堆排序初始建堆后结果。

5.无向图的邻接矩阵如下所示, 顶点序列为{a,b,c,d,e,f}, 试用 Prim 算法从顶点 a 开始求该图的最小生成树, 并给出依次产生的边, 边用(x,y)的形式表示。

$\infty$	6	1	5	$\infty$	$\infty$
6	$\infty$	5	$\infty$	3	$\infty$
1	5	$\infty$	5	$\infty$	4
5	$\infty$	5	$\infty$	$\infty$	2
$\infty$	3	$\infty$	$\infty$	$\infty$	6
$\infty$	$\infty$	4	2	6	$\infty$

6.{45, 80, 48, 40, 22, 78}, 构造平衡二叉树

7.求下图的关键路径，要求计算出事件和活动的最早开始时间和最晚开始时间。



8.在包含快表的段页式存储管理中，访问一个变量要进行多少次内存访问？

9.逻辑地址长度位 16 位，其中高 6 位表示页号，低 10 位表示页内偏移量，试回答下列问题：

- (1) 逻辑地址有多少页？页面大小为多少？
- (2) 可存放的文件最大长度为多少？

10.采用混合索引分配方式的某文件系统中，索引结点中共有 13 个地址项：10 个直接地址项，1 个一次间址项，1 个二次间址项和一个三次间址项。每个盘块的大小为 4KB。若每个盘块号需要用 4 个字节来描述，则对某个长度为 8192001 个字节的文件，需要占用多少个盘块？(注意，盘块计算结果中不包括 13 个地址项所在的索引结点的占用空间)

11.现有一请求分页的虚拟存储器，内存最多容纳 5 个页面，用 FIFO，LRU，改进型 CLOCK 页面置换算法分别指明最先淘汰哪一页，并说明理由。

	装入时间	上次访问时间	读位	修改位
0	50	69	0	0
1	7	120	1	0
2	2	110	1	1
3	66	100	1	1
4	15	77	0	1

12.根据文件的物理存储结构可将文件分为哪几类，请简述。

四、编程题（每题 15 分，共 30 分）

1.在带头结点的单链表中，设计一个算法删除值重复的结点。

2.查找二叉树中给定值为  $k$  的结点的双亲结点。

上海海电大学计算机考研交流群: 655764640



## 2024 年上海海事大学攻读硕士学位研究生入学考试试题答案

(重要提示：答案必须做在答题纸上，做在试题上不给分)

考试科目： 828      考试科目名称： 计算机综合

### 一、选择题

1.C	2.B	3.C	4.B	5.A
6.C	7.B	8.A	9.D	10.B
11.D	12.A	13.A	14.B	15.B

1.C

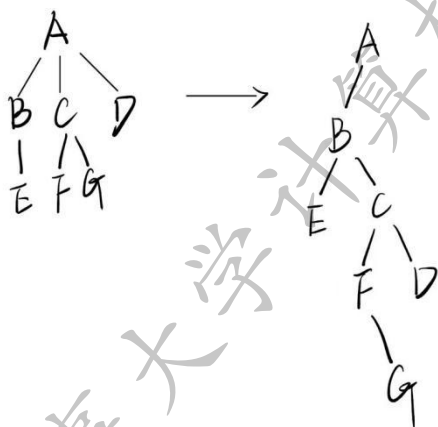
解析：对称矩阵进行压缩存储，只需要存放上三角或者下三角区域的数据元素，那么有  $1+2+3+\dots+10=55$  个元素，所以答案为 C。

2.B

解析：通常一堆元素中找其中最大或者最小的  $k$  个元素选择堆排序。

3.C

解析：这题考察树与二叉树的转换，记住左孩子右兄弟的口诀即可，转换后的二叉树如下图右边所示，高度为 5。



4.B

解析：注意这里是保证该图连通，而不是  $m$  个顶点的连通图至少要有  $m-1$  条边，也就是无论你的边是怎么连，都要保证该图是连通图，所以前  $m-1$  个顶点组成一个完全图，再加入任意一条边则变成连通图，所以是  $(m-1)(m-2)/2+1$ 。

5.A

解析：按序号查找就是访问第  $i$  个元素，仍然是随机存取，所以为顺序存储。比

如，访问第六号元素。

6.C

解析：以下内部排序方法中时间复杂度既为  $O(n\log n)$  且稳定的只有归并排序。

7.B

解析：选项 A，5 出栈后，此时栈内还剩 1 2 3 4（由左往右靠近栈顶），那么 3 不可能在 4 之前出栈。对于选项 C，3 出栈后，此时栈内还剩 1 2（由左往右靠近栈顶），那么 1 不可能在 2 之前出栈。对于选项 D，1 入栈随后出栈，2 3 4 5 依次入栈，5 4 再出栈，6 入栈再出栈，此时栈内还剩 2 3，那么 2 不可能在 3 之前出栈。

8.A

解析：简单模式匹配算法的最坏时间复杂度是  $O(nm)$ ，其中  $n$  和  $m$  分别是主串和模式串的长度。

9.D

解析：快速排序在划分过程中,用到递归算法,其递归算法平均深度为约为  $\log_2 n$ ，所以其空间复杂度为  $O(\log_2 n)$ 。

10.B

解析：对于无向图边表的结点数是无向图中边数的两倍，则图中的边数应为  $m/2$ 。

11.D

解析：并发进程失去封闭性是指多个程序共享系统资源，因而资源的状态将由多个程序来改变。

12.A

解析：死锁的四个必要条件中不能被破坏的是互斥条件。

13.A

解析：SPOOLing 将独占设备改造为共享设备，但不是共享分配，而是通过加了一个虚拟层实现了从物理设备到逻辑设备的映射，从而实现了虚拟分配。

14.B

解析：快表是一种高速缓冲存储器，不在内存。

15.B

解析：银行家算法用来避免死锁的。

## 二、填空题

### 1.正确性

解析：算法设计的要求：正确性、可读性、健壮性、效率与低存储量需求

### 2. $(Q.rear - Q.front + n) \% n$

解析：该题是求循环队列的长度： $(Q.rear - Q.front + \text{MaxSize}) \% \text{MaxSize}$ ，则答案为  $(Q.rear - Q.front + n) \% n$ 。

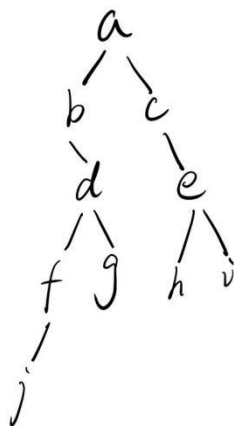
### 3. {45, 80, 40, 22, 48, 78}

解析：直接插入排序步骤：

- ①从序列第一个元素开始，该元素可以认为已经被排序；
- ②取出下一个元素，设为待插入元素，在已经排序的元素序列中从后向前扫描，如果该元素（已排序）大于待插入元素，将该元素移到下一位置；
- ③重复步骤 2，直到找到已排序的元素小于或者等于待排序元素的位置，插入元素；
- ④重复 2, 3 步骤，完成排序。

### 4. bjfdgachei

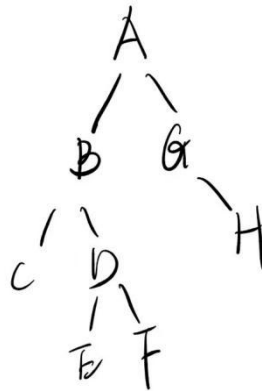
解析：由顺序存储画出该二叉树的图只需要记住对于第  $i$  个结点，第  $2i$  个结点是其左孩子，第  $2i+1$  个结点是其右孩子。二叉树如下图所示，则中序序列为 bjfdgachei。



### 5. ABCDEFGH

解析：由中序序列和后序序列可得该二叉树，因此该二叉树的先序序列为

ABCDEFGH。



6.3

解析：广义表的长度指表中元素的个数，即 3。

7.  $O(\log_5 n)$

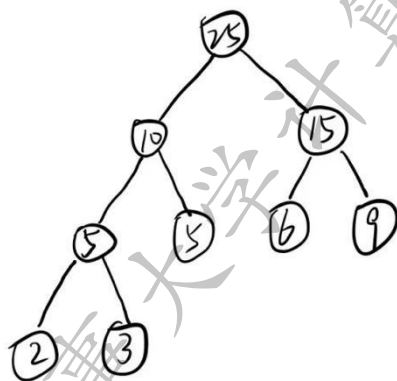
解析：设循环次数为  $t$ ，则

$t$	0	1	2	3	...	$t$
$i$	0	5	$5^2$	$5^3$	...	$5^t$

所以有  $5^t > n$ ， $t = \log_5 n$ ，所以时间复杂度为  $O(\log_5 n)$

8.55

解析：WPL =  $(2+3)*3 + 5*2 + (6+9)*2 = 55$



9.19

解析：

后缀表达式的求值：从左→右扫描，遇到运算符将该运算符前面两个数取出运算，合体为一个操作数，结果带回后缀表达式参与后面的运算，直到扫描到最后一个运算符并计算得最终结果。

所以  $abc-/de*+$  转为中缀表达式为  $a/(b-c)+(d*e)$ ，即  $6/(5-3)+(2*8)=19$

10.2n

解析：n 个叶子结点的哈夫曼树会新建 n-1 个新结点，则共有 2n-1 个结点，所以二叉链表中 2n-2 个有效的指针域，2n 个空指针域。

11.15

解析：7 个并发进程，每个进程需要资源最大为 3 个。极端情况，每个进程已经分配 2 个资源，此时其中任何一个进程只需要再分配一个资源就能满足它的最大需求，进程总能执行下去，所以为  $2*7+1=15$ 。

12.[1-N,1]

解析：当没有进程使用时是 1 且最多是 1。当有进程申请时便减 1 第一个申请是 0，第二个是 -1，...，第 N 个是 1-N。

13. 短作业优先

解析：短作业优先调度算法平均等待时间，平均周转时间最少。

14. 寻道时间

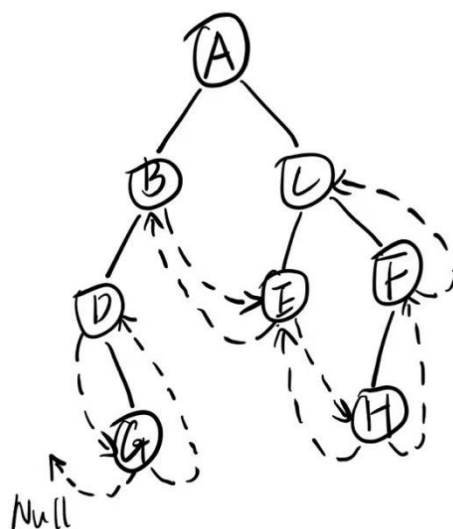
解析：磁盘访问时间由寻道时间、旋转延迟时间、传输时间组成。

15. 程序接口（系统调用）

解析：OS 的接口包括命令接口，图形界面接口和程序接口。

### 三、应用题

1. 该二叉树的后序序列为 GDBEHFCA，所以后序线索二叉树如下图所示。



2.解析:

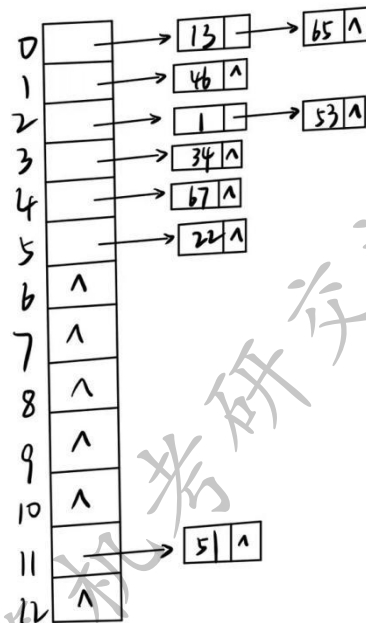
$$H(1) = 2 * 1 \% 13 = 2 \quad H(13) = 2 * 13 \% 13 = 0$$

$$H(51) = 2 * 51 \% 13 = 11 \quad H(46) = 2 * 46 \% 13 = 1$$

$$H(22) = 2 * 22 \% 13 = 5 \quad H(53) = 2 * 53 \% 13 = 2$$

$$H(67) = 2 * 67 \% 13 = 4 \quad H(65) = 2 * 65 \% 13 = 0$$

$$H(34) = 2 * 34 \% 13 = 3$$



3.

解析: 森林转换成二叉树的画法如下

- ①将森林中的每棵树转换成相应的二叉树;
- ②每棵树的根结点视为兄弟关系, 在每棵树的根之间加一根连线;
- ③以第一棵树的根为轴心顺时针旋转 45°。

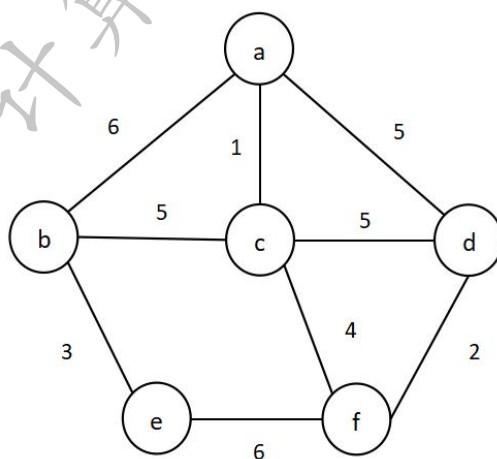


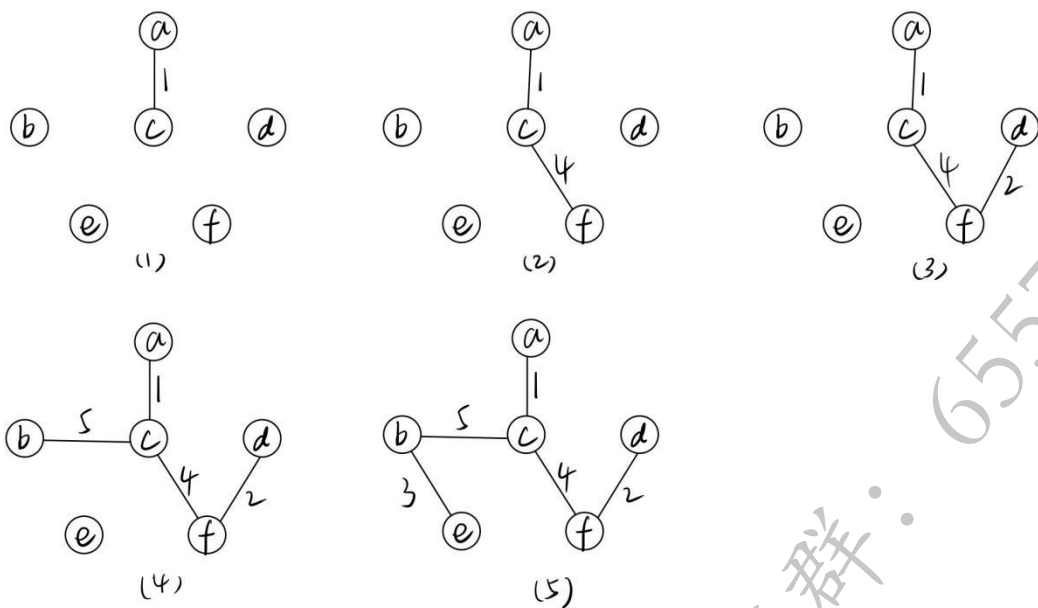
4.解析:

(1) 快速排序第一趟结果: 20 6 29 27 15 32 92 97 50

(2) 因为是递增, 所以为大根堆。初始建堆结果为: 97 32 92 29 6 15 50  
27 20

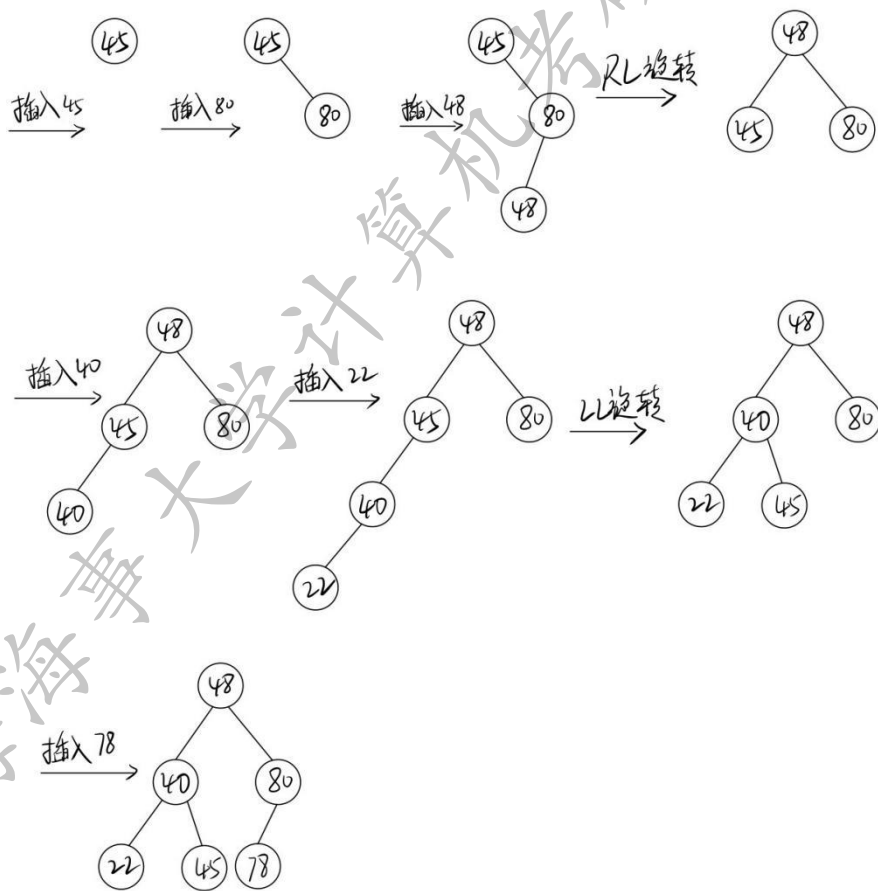
5.根据邻接矩阵可画出该图, 如下图所示





依次产生的边为(a,c), (c,f), (f,d), (b,c), (b,e)。

6.平衡二叉树构造过程如下所示





7.

	V0	V1	V2	V3	V4	V5
ve(i)	0	3	2	6	6	8
vl(i)	0	3	2	6	6	8

	a1	a2	a3	a4	a5	a6	a7	a8
e(i)	0	0	3	3	2	6	6	2
l(i)	0	0	3	4	2	6	6	5
l(i)-e(i)	0	0	0	1	0	0	0	3

所以关键路径为(V0, V1, V4, V5)和(V0, V2, V3, V5)。

8.

解析：在包含快表的段页式存储管理中，在地址变换机构中增设一个高速缓冲寄存器。每次访问时，同时利用“段号+页号”去检索高速缓存，若找到匹配的表项，可从中得到相应的物理块号，与页内地址一起形成物理地址，此时只需访问一次内存；若未找到匹配表项，则仍须再三次访问内存。

9.解析：

(1) 高 6 位表示页号，低 10 位表示页内偏移量，则进程最多有 $2^6$ 个页面，页面大小为 $2^{10}\text{B}$ 。

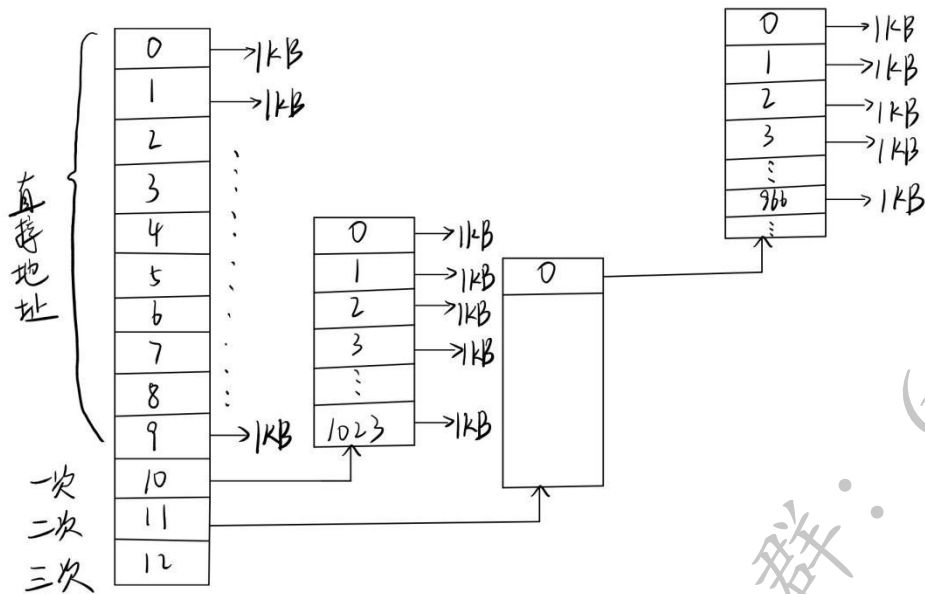
(2) 可存放的文件最大长度为 $2^6 \times 2^{10} = 2^{16}\text{B}$ 。

10.解析：与 23 真题类似。

$4\text{KB}/4\text{B}=1024$  所以一个盘块最多存储 1024 个块号

由文件长度和磁盘块大小可知，文件需要  $8192001/4096=2001$  个磁盘块

直接地址项直接对应所指的磁盘块，可记录 10 个磁盘块；一次间址项使用 1 个间接索引块，共可记录 1024 个磁盘块；现在还剩余  $(2001-10-1024)=967$  个磁盘块，二次间址项使用  $1 + \lceil 967/1024 \rceil = 2$  个间接索引块，所以占用  $2001+1+2=2004$  个盘块，具体方法如图所示。



11.解析:

FIFO 算法

最先淘汰第 2 页，因为 FIFO 优先淘汰最早进入内存的页面。

LRU 算法

最先淘汰第 0 页，因为 LRU 选择最近最长时间未访问的页面进行淘汰。

改进 CLOCK 算法

最先淘汰第 0 页，因为改进 CLOCK 优先选择读位=0，修改位=0 的页面进行替换。

12.解析：根据文件的物理存储结构，文件可以分为顺序文件、链接文件和索引文件三类。

一、顺序结构（连续结构）

定义：顺序结构是一种最简单的物理文件结构，它将逻辑上连续的文件信息依次存放在连续编号的物理块中。

优点：顺序存取速度快，顺序访问容易。

缺点：要求有连续的存储空间。

二、链接结构

定义：文件分配到离散的盘块，通过链接指针将这些离散的盘块链成一个链表。

特点:

优点: 消除了外存的碎片, 提高了外存的利用率, 并使文件很容易实现动态增长。

缺点: 不适合随机访问, 因为需要按照链指针的顺序逐个访问物理块。

### 三、索引结构

定义: 索引结构将文件存放在外存的若干个离散的盘块中, 并为每个文件建立一个索引表。索引表中的每个表目存放文件信息所在的逻辑块号和与之对应的物理块号。

优点: 访问速度快, 因为可以直接通过索引表找到所需的物理块。同时, 文件长度可以动态变化, 因为物理块不需要连续存放。

缺点: 存储开销大, 因为每个文件都需要一个索引表, 而索引表也需要物理块来存储。此外, 当文件被打开时, 索引表需要读入内存, 占用额外的内存空间。

### 四、编程题

1.

// 链表结点定义

```
typedef struct node {
```

```
    float data;
```

```
    struct node *next;
```

```
} Node, *List;
```

List Delete (List &L) { //删除带头结点的单链表 L 中所有重复元素

```
    p=L->next; //p 用于遍历链表
```

```
    while (p!=NULL) {
```

```
        pre=p; //pre 记录当前考察结点的前驱结点
```

```
        q=p->next; //q 表示当前考察结点, 若与 p 的 data 相同则删除
```

```
        while (q!=NULL) {
```

```
            if (q->data==p->data) {
```

```
                pre->next=q->next;
```

```
                free(q);
```

```
                q=pre->next;
```

```
            }
```

```

        else{
            pre=q;
            q=q->next;
        }
    }
    p=p->next;
}
return L;
}

```

2.

// 二叉树结点定义

```
typedef struct BiTNode{
```

```
    int data;
```

```
    struct BiTNode * lchild,*rchild;
```

```
}BiTNode, *BiTree;
```

```
BiTree findParent(BiTree T, int k) {
```

```
    if (T == NULL || T->data == k)
```

```
        return NULL;
```

```
    if (T->lchild && T->lchild->data == k || T->rchild && T->rchild->data == k)
```

```
        return T; // 如果 T 的左子节点或右子节点的值等于 k，则 T 是 k 的父节
```

点

```
    // 递归地在左子树和右子树中查找
```

```
    BiTree parent = findParent(T->lchild, k);
```

```
    if (parent == NULL)
```

```
        parent = findParent(T->rchild, k);
```

```
    return parent;
```

```
}
```