

复 旦 大 学

2003 年招收攻读硕士学位研究生入学考试试题

考试科目: 数据结构与操作系统

注意:答案请做在答卷纸上,做试题上一律无效。

(共 6 页)

(满分 150 分)

一、填空题 (本大题共四小题, 10 空, 每空 2 分, 共 20 分)

1. 操作系统两个最基本的特征是 (1) 和 (2), 两者之间互为存在条件。
2. 在多道程序系统中, 进程之间存在着的不同关系可以划分为两类: (1) 指进程间存在一定的逻辑关联, 需要在执行次序上协调; (2) 指进程间在使用共享资源方面的约束关系。解决上述两个问题的一种常用方法是 (3)。
3. 某进程页面访问序列为 4, 3, 2, 1, 4, 3, 5, 4, 3, 2, 1, 5 且开始执行时内存中没有页面, 分配给该进程的物理块数是 3。则采用 FIFO 页面置换算法时缺页率是 (1), 采用 LRU (最近最久未使用) 页面置换算法时缺页率是 (2)。
4. 虚拟设备是通过 (1) 技术把 (2) 设备变为能为若干个用户 (3) 的设备。

二、简答题 (本大题共二小题, 共 15 分)

1. 为什么说多级反馈调度算法能较好地满足交互类、短批处理类和长批处理类等应用的进程的需要? (本题 8 分)
2. 请指出操作系统的系统调用与一般过程调用的主要区别。(本题 7 分)

三、综合题 (本题 15 分)

在某系统中, 数据从磁盘读入缓冲区, 然后从缓冲区传入用户区, 再在用户区中处理。假设该磁盘系统中, 文件在磁道上非连续存放, 磁头从一个磁道移至另一个磁道需要时间 t_1 , 逻辑上相邻数据块的平均距离为 d 磁道, 每块的旋转延迟时间及传输到缓冲区的传输时间分别为 t_2 和 t_3 。问读取 N 个数据块的磁盘访问时间一共是多少? 另外, 假设将缓冲区的数据传送到用户区所花费的时间为 t_4 且 t_4 远远小于读取一个数据块的磁盘访问时间, CPU 对一块数据进行处理的时间为 t_5 。问分别在单缓冲和双缓冲情况下, 一块数据的总处理时间为多少?

四、填空题 (本大题共 8 小题, 每小题 5 分, 共 40 分)

1. 设环形队列存放在数组 q 中, 数组 q 的长度为 L , 下标从 $0 \sim n-1$, 队头指针 $head$ 指向队头结点, 队尾指针 $tail$ 指向队尾结点后一个空闲结点。
该环形队列的队空标志为 (1), 队满标志为 (2),
该队列的长度为 (3)。
2. 设数组 $a[n][n]$ 是一个二维数组, 数组中存放的是下三角矩阵。所谓下三角矩阵是指主对角线的右上方元素都是零的方阵。现将 a 中下三角的元素 (包括主对角线) 按行优先存储于一维数组 b 中。 (数组 a 、 b 中下标从 0 开始计)
设数组 a 中某下三角元素为 $a[i][j]$, 存放在数组 $b[k]$ 中, 则 $k =$ (4)。
如果 a 中下三角元素按列优先存储在一维数组中, 则 $k =$ (5)。
3. 设有算术表达式 $x+a*(y-b)-c/d$, 该表达式的前缀表示为 (6),
后缀表示为 (7)。
4. 对关键码序列 FBJGEAIDCH 进行升序排列, 则堆排序时, 初始建堆结果的序列为 (8)。设关键码序列中有 n 个元素, 则堆排序的平均执行时间和需附加的存储结点分别为 (9)。
5. 在树转换成二叉树形式时, 二叉树中每个结点的左子结点是它原来 (树中) 的 (10) 结点, 二叉树中每个结点的右子结点是它原来 (树中) 的 (11) 结点。
6. 有 n 个顶点的有向强连通图最多有 (12) 条边, 最少有 (13) 条边。
有 n 个顶点的无向连通图最多有 (14) 条边, 最少有 (15) 条边。
7. 在 m 阶 B 树中, 除了根和叶子外, 每个结点的子结点数目的范围在 (16) 之间, 同时, 具有 k 个子结点的非叶子结点含有 (17) 个键值。
而在 m 阶 B+树中, 具有 k 个子结点的结点含有 (18) 个键值。
8. G 是一个非连通无向图, 共有 28 条边, 则该图至少有 (19) 个顶点。

五. 下面是一个链接存储线性表（带“表头结点”）的选择排序函数。排序过程中，依次选出键值从小到大的结点可顺序组织成有序链表，即每次选出的键值最小的结点接在已部分完成的有序链表的末尾。请在空框处填上适当内容，每个空框只填一个语句。（本题 10 分）

```
typedef struct node { char data;
                      struct node *link;
                    } NODE;
NODE *select_sort (NODE *h)
{ NODE *tail, *u, *v, *p;
  tail = h;
  while (tail->link != NULL)
  { for (u = tail->link, p = tail; u->link != NULL; (1) )
    if (u->link->data < p->link->data) (2) ;
    if (p != tail)
    { v = p->link; p->link = v->link;
      (3) ; (4) ;
    }
    (5) ;
  }
  return h;
}
```

六. 已知一棵二叉树的前序遍历的结点序列为 ABCDEFG,

中序遍历的结点序列为 BDCAGFE,

试画出这棵二叉树，并写出后序遍历时得到的结点序列。（本题 10 分）

七. 下面的递归函数是根据二叉树的前序遍历结果和中序遍历结果, 产生一棵二叉树。前序遍历结果在形参 pstr 中, 中序遍历结果在形参 istr 中。产生的二叉树作为函数值返回。试在空框处填上适当内容。 (本题 10 分)

```
typedef struct node { char data;
```

```
    struct node *lchild;
```

```
    struct node *rchild;
```

```
    } NODE;
```

```
NODE *restore (char *pstr, char *istr, int n)
```

```
{ NODE *ptr;
```

```
  char *rstr;
```

```
  int k;
```

```
  if (n <= 0) return NULL; /*n 为字符串长度*/
```

```
  ptr = (NODE *) malloc (sizeof (NODE));
```

```
  ptr -> data = *pstr;
```

```
  for (rstr = istr; rstr < istr + n; rstr++)
```

```
    if (*rstr == *pstr) (1); /*在中序遍历中找根结点*/
```

```
  k = (2); /*左子树的结点数*/
```

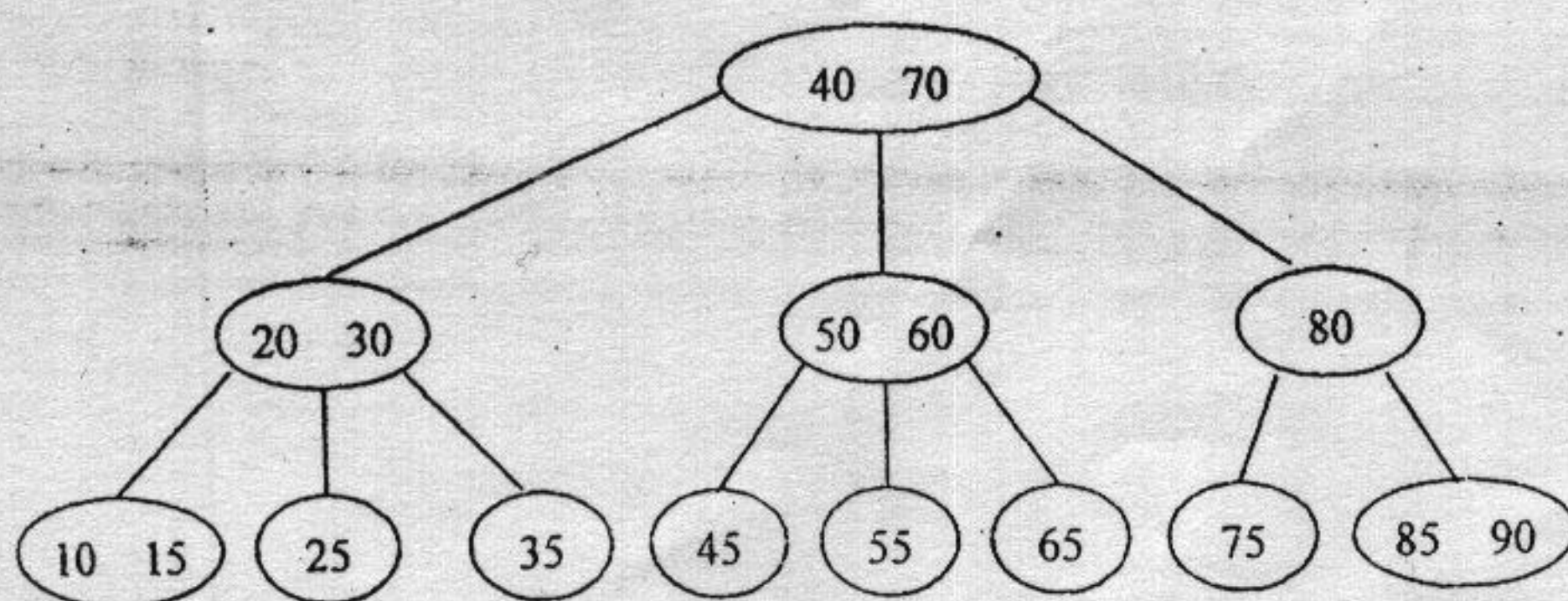
```
  ptr -> lchild = restore ((3));
```

```
  ptr -> rchild = restore ((4));
```

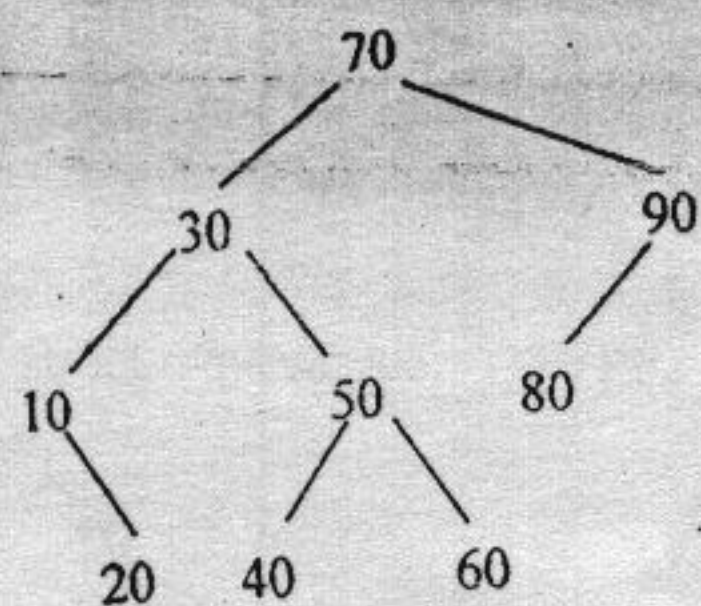
```
  return ptr;
```

```
}
```

八. 下图为 3 阶 B 树的简图, 若在此树上依次插入关键字 88、12, 试分别画出 B 树的变换过程。 (本题 10 分)

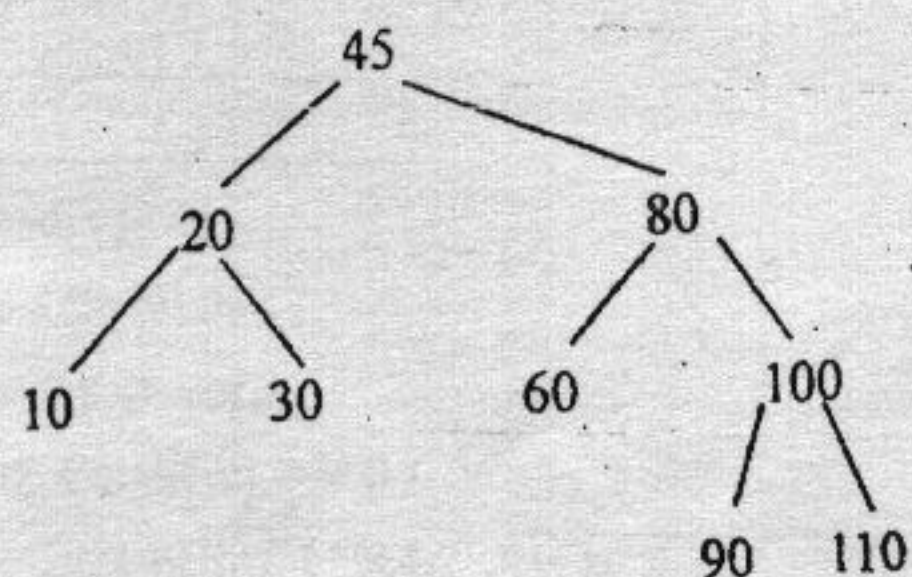


九. 下面给出两棵平衡查找树及相应的插入键值。试分别用 Adelson 插入算法
(本题 10 分)
执行相应插入后的平衡查找树。



插入 55 →

答案画在答题纸上。



插入 95 →

答案画在答题纸上。

十. 下面是用广度优先搜索法遍历图的函数。请在空框处填上适当内容, 每个空框只填一个语句。

(本题 10 分)

```
#define MAXN 50
typedef struct l_node {int ver;
                      struct l_node *link;
                      }L_NODE;
L_NODE *head[MAXN]; /*邻接表*/
int visit[MAXN];

void bfs (u)
int u;
{struct queueType { int qa, qe; /*队列首指针、尾指针*/
                  int item[MAXN]; /*队列数组*/
                  };
typedef struct queueType QTYPE;
int v, w;
L_NODE *t;
QTYPE queue;
printf ("%4d", u); visit[u] = 1;
queue.qa = 0; queue.qe = 0; queue.item[0]=u;
while ( (1) )
{ v = (2);
  t = head[v];
  while (t!=NULL)
  { w = t->ver;
    if (visit[w]==0)
    { printf ("%4d", w);
      visit[w]=1;
      (3);
    }
    (4);
  }
}
}
```