

北京大学
2005 年硕士研究生入学考试试题

考试科目：计算机软件基础

考试时间：2005 年 1 月 23 日下午

招生专业：计算机软件与理论、计算机应用

一、(35 分)基础知识填空

1. 对于非空满 K 叉树，其分支结点数目为 n ，则其叶结点数目为_____。
2. 高度为 h 的 AVL 树上的最少结点个数是_____个，最多结点个数是_____。
3. 有 n 个顶点的有向连通图最最少有_____条边。
4. 某二叉树中序序列为 A, B, C, D, E, F, G, 后序序列为 B, D, C, A, F, G, E, 则前序序列是_____。
5. 关键码序列 (tang, deng, an, wan, shi, bai, fang, liu) 按字典顺序排序 (“an” < “bai”, 空格小于任何其它字符)。采用“右补空格”，使得这些关键码都成为 4 位字符的等长关键码。按最低位优先法进行基数排序，进行两趟分配和收集后得到的序列为_____。
6. 将一个 $A[1..100, 1..100]$ 的三对角矩阵，按行优先顺序存储在一维数组 $B[1..298]$ 中， A 中元素 $A[66][65]$ 在 B 数组中的位置 K 为_____。

注释：三对角矩阵是对角线以及对角线的上下两条与对角线斜率相同的紧邻斜线上的元素不为零，而其他位置元素都是零的矩阵。它的特点是第一行和最后一行各有两个非零元素，其他各行各有三个非零元素。

二、(30 分)辨析题

1. 请设计一种“自调整”数据结构。假设检索的关键码是中文词组。要求：
 - (1) 定义其 ADT (说明其逻辑结构，以及主要的运算)。
 - (2) 简单地说明它怎样根据所检索过的中文词组的而进行调整，使得最近出现的词组能比较快速地被检索到。
2. 请判断下述计算前缀表达式的算法是否正确。
 - (a) 如果正确，请给出一个简单实例，演示算法运行步骤；
 - (b) 如果不正确，请指出错误之处，并给出改正后的算法。如果只是局部修改，可以只具体给出所修改的部分；其他未修改部分，用其原来的功能编号(1)(2)、a、b.1、b.2 指代。

【计算前缀表达式的算法】

// 利用一个栈，从左向右扫描，边扫描边求值。

(1) 从左向右扫描前缀表达式：

a: 当遇到的是一个运算符时，则操作数入栈，转到 1。

b: 对于操作数，如果栈为空，则将该操作数返回，退出；如果栈不空：

b.1: 当栈顶是一个运算符时，则将操作数压入栈中，转到 1；

b.2: 当栈顶也是一个操作数时，两次取栈顶 (第二次取得的应为操作符)，并按照运算符对两个操作数进行计算 (其中从栈中取得的操作数为第一操作数)，所得计算结果作为操作数，将新操作数入栈，转到 1。

(2) 扫描结束，正确的情况下，输出结果应该在 b 处返回，并退出；如果程序没有在 b 处返回退出，而是进行到此处，则表明出错，输出错误消息。

【计算前缀表达式的算法结束】

注释：前缀表达式与后缀表达式一样，不包含括号，运算符放在两个运算对象的前面，如： $*+213$ 。

3. 请判断下面快速模式匹配算法 KMP-FindPat0 是否正确。

(a) 如果正确，请给出一个简单实例，演示算法运行步骤；

(b) 如果不正确，请指出错误之处，并给出改正后的算法。如果只是局部修改，只需要具体给出所修改的部分（例如，第 x 行 - 第 y 行代码修改为……）。

```
int * Next(String P) {
int m = P. strlen( );           //m 为模板 P 的长度
assert( m > 0 );                 //若 m=0, 退出
int * N = new int[ m ];         //动态存储区开辟整数数组
assert( N != 0 );               //若开辟存储区域失败, 退出
N[0] = 0;
//分析 P 的每个位置 i
for( int i = 1 ; i < m ; i + + ) {
    int k = N[i - 1];           //第(i-1)位置的最长前缀串长度
    //以下 while 语句递推决定合适的前缀位置 k
    while( k > 0 && P[i] != P[k] )
        k = N[k - 1];
    //根据 P[i] 比较第 k 位置前缀字符, 决定 N[i]
    if( P[i] == P[k] )
        N[i] = k + 1;
    else N[i] = 0 ;
}
return N;
}

int KMP_ FindPat(String S, String P, int * N, int startindex) {

    //假定事先已经计算出 P 的特征数组 N, 并作为输入参数
    int LastIndex = S. strlen( ) - P. strlen( );
    if ( ( LastIndex - startindex ) < 0 )
        return( -1 );           //匹配失败成功
    int i;                       //i 是指向 S 内部字符的游标
    int j = 0;                   //j 是指向 P 内部字符的游标
    for(i = startindex; i < LastIndex; i + + ) { //i 是 S 游标

        //若当前位置的字符不同, 则用 N 循环求当前的 j
        While ( p[j] != S[i] )
            j = N[j - 1];        //寻找 j 将 P 的恰当位置与 S 的 i 位置对准

        if ( P[j] == S[i] ) //P[j] == S[i] 位相同时, 则继续下一步循环

            j + + ;
        //若当前位置的字符不同, j 将由下一步循环的 N 循环决定,

        if ( j == P. strlen( ) )
            return(i - j);       //匹配成功, 返回该 S 子串的开始位置

    }
    return( -1 );               //P 和 S 整个匹配失败
}
```

三、(15 分) 算法填空

下面是一个拓扑排序算法，对于带回路的图，该算法将调用“此图有环！”，并逐步退出

到调用它的主调函数。请填充算法中空缺的部分，使其成为完整的算法。

//深度优先搜索方式实现拓扑排序

```
void Do_topsort_ Circle(Graph & G, int V, int * result, int & tag)
{
    if( G. Mark[ V] == VISITED) {
        空缺 1

    }
    G. Mark[ V] = VISITED;
    for( Edge e = G. FirstEdge( V); G. Is Edge(e); e = G. Next Edge(e)) {
        if( 空缺 2 )
            Do_topsort_ Circle( G, G. To Vertex(e), result, tag);
    }
    result[ tag + + ] = V;
    空缺 3
}
```

四、(40 分)问答题

1. 实现进程通信的方式有几种？请分别简要描述这些通信方式。
2. 在实现虚拟页式存储管理方案时，页表表项是由什么决定的？通常页表设置哪些表项？每一表项的作用是什么？
3. 文件系统的性能对整体系统的性能影响很大，请说明在实现文件系统时可以从哪些方面提高文件系统的性能，简要给出这些手段的具体解决思路。

五、(20 分)综合应用题

1. 系统中有 5 个进程，每个进程的运行时间(单位：ms)、优先级和到达时刻如下表所示：

进 程：	P1	P2	P3	P4	P5
运行时间：	10	1	2	1	5
优 先 级：	4	6	2	3	6
到达时刻：	0	1	2	3	4

请给出当系统分别采用时间片轮转算法(时间片为 1ms)、不可抢占优先级调度算法和抢占式优先级调度算法时，各进程的执行情况。

2. 假设有如下访盘请求，请计算出对这些请求的服务次序，使得平均访问时间最短。设当前磁头的位置是 6 号柱面。

请求顺序	柱面号	磁头号	扇区号
①	3	2	1
②	5	1	5
③	3	2	5
④	3	4	1
⑤	9	2	1
⑥	9	1	5
⑦	5	2	5
⑧	5	4	8

六、(10 分)P、V 操作题

某系统如此定义 P、V 操作：

P(S)

$S = S - 1;$

若 $S < 0$, 本进程进入 S 信号量等待队列的末尾; 否则, 继续执行。

V(S)

$S = S + 1;$

若 $S \leq 0$, 释放等待队列中末尾的进程, 否则继续运行。

现有四个进程 P_1 、 P_2 、 P_3 、 P_4 竞争使用某一个互斥资源(每个进程可能反复使用多次), 试用上面定义的 P、V 操作正确解决 P_1 、 P_2 、 P_3 、 P_4 对该互斥资源的使用问题。