

## 98 届数据结构期终考卷(A 卷)

一 根据下列各题要求填空(每空 3 分, 共 15 分):

1. 一个非空二叉树的叶节点的个数 ( $n_0$ ) 和 有二个分枝的节点的个数 ( $n_2$ ) 的关系为:  
A)  $n_0 = n_2 - 1$  B)  $n_0 = n_2 + 1$  C)  $n_2 = n_0 - 1$  D)  $n_2 = n_0 + 1$
2. 对哈希(HASH)函数  $H(k) = k \text{ MOD } m$ , 一般来说,  $m$  应取  
A) 素数 B) 很大的数 C) 偶数 D) 奇数
3. 设计一个判断表达式中左右刮号是否配对的算法, 采用\_\_\_\_\_数据结构最佳.  
A) 队列 B) 堆栈 C) 二叉树 D) 链表
4. 一个  $n$  个顶点的连通无向图, 其边的个数至少为:  
A)  $n-1$  B)  $n$  C)  $n+1$  D)  $n \log n$
5. 已知一棵二叉树的前序遍历结果为 ABCDEF, 中序遍历结果为 CBAEDF, 则后序遍历的结果为:  
A) CBEFDA B) FEDCBA C) CBEDFA D) 不定

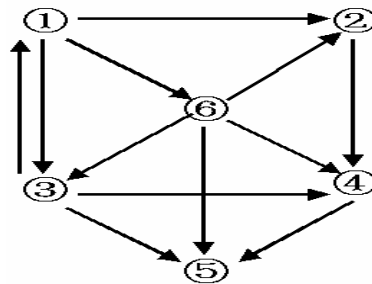
二、考察一个二处理机的调度问题, 各作业所需的不同处理机的时间由矩陈 J 给出

$$J = \begin{bmatrix} 3 & 6 \\ 4 & 2 \\ 8 & 9 \\ 10 & 15 \end{bmatrix} \quad \begin{matrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{matrix}$$

根据动态规划法进行最优调度, 写出其四个作业的处理次序及完成所有作业的时间 (6 分)

三. 右图所示为一有向图, 请给出该图的下述要求:

- 1) 每个顶点的入/出度。
- 2) 邻接矩阵。
- 3) 邻接表。
- 4) 逆邻接表。
- 5) 强连通分量。 (15 分)



四. 广义表的结点结构如右图所示:

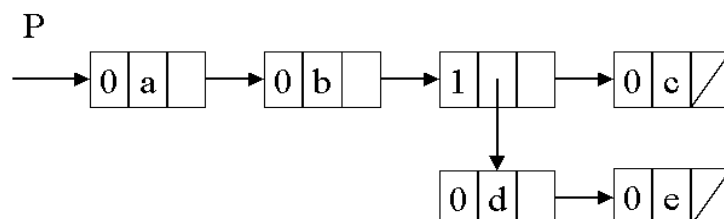
其中 LINK 为表中指向下一元素的指针,

TAG	DATA	LINK
-----	------	------

TAG 为标志域, TAG=0 表示该结点为原子, DATA 为数据; TAG=1

表示该结点为一子表, DATA 为指向子表的指针。下图为一广义表的例子。

试用 C 语言编写一程序, reverse(p,t),

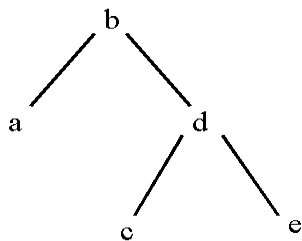


将 p 所指的广义表倒置(包括所有子表), 即将所有 LINK 指针逆转, t 为指向新表表头的指针。(12 分)

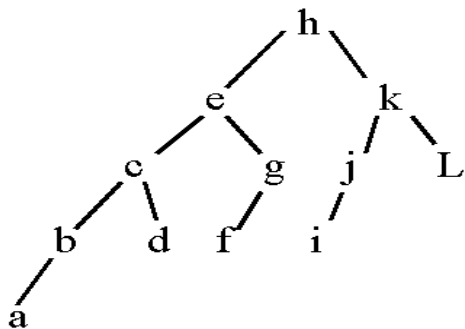
五. 要在 10000 个整数中找出 10 个最大的数, 和 10 个最小的数, 选择哪种数据结构和哪种 Sorting 算法最好, 为什么? 写出实现该功能的算法。(12 分)

六. (a)试画出在图(a)的 AVL 树中插入新结点 f 后的结果。  
(b)试画出在图(b)的 AVL 树中删去结点 L 后的结果。(10 分)

(a)



(b)



七. 阅读如下两个程序, 指出其功能, 并在空格处填上适当语句。(12 分)

```

1) void sub1(element list[], int root, int n)
{
    int child, rootkey;
    element temp;
    temp = list[root];
    rootkey = list[root].key;
    child = 2 * root;
    while ( child <= n) {
        if ((child < n) &&
            (list[child].key < list[child+1].key))
            child++;
        if (rootkey > list[child].key)
            _____;
        else { _____;
                child *= 2;
            }
    }
    list[child / 2] = temp;
}
  
```

```

void test1(element list[], int n)
{
    int i, j;
    element temp;
    for (i = n/2; i > 0; i--)
        sub1(list, i, n);
    for (i = n-1; i > 0; i--) {
        swap(list[1], list[i+1], temp);
        sub1(list, 1, i);
    }
}

```

```

2) void test2(element item, list_pointer ht[])
{
    int hash_value = hash( item.key) ;
    list_pointer ptr, trail = NULL, lead = ht[hash_value];
    for ( ; lead; trail = lead, lead = lead->link) {
        if (!strcmp(lead->item.key, item.key)) {
            fprintf(stderr, "The key is in the table\n");
            exit(1);
        }
    }
    ptr = (list_pointer)malloc(sizeof(list));
    if (IS_FULL(ptr)) {
        fprintf(stderr, "The memory is full\n");
        exit(1);
    }
    ptr->item = item;
    ptr->link = NULL;
    if (trail)
        _____ ;
    else
        _____ ;
}

```

八. 右图为一 AOE(activity on Edge)图。

1. 列出每一顶点的 earlist 和 latest 时间。
2. 指出关键路径 (critical path)
3. 写出计算 earlist time 的算法。(18 分)

