2016年哈工大计算机科学与技术专业期末题解析

一、单项选择题（每小题2分，共20分）

答案概览:

1-5: A B A D B	6-10: C B C B C

详细解答:

1、选A

我们可以很容易列出式子

$\frac{n^2}{2^k}=1$

$k=2log2n$

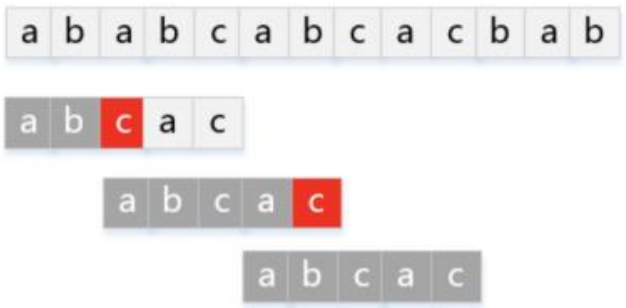
因此我们得出结果为A

2、选B

插入和删除操作不需要移动元素且需要分配一个较大的存储空间只有B满足要求，对于A单向链表虽然插入和删除不需要移动元素，但是也不需要分配较大的空间，对于C和D而言删除和插入操作都需要O（n）的时间复杂度。

3、选A

模式串T="abcac"匹配主串S="ababcabcacbab"的KMP过程如下图：



因此，得到最终的匹配结果需要 3 次。

其中：子串的 next 数组为：[-1, 0, 0, 0, 1]，当然算法具体实现的差异，也有使用 next = [0, 1, 1, 1, 2]，只是起始点的比较不一样，没有差异。

4、选 D

根据前序遍历我们可以知道 1 为根结点，对于 D 选项 3 为 1 的左子树这和题目中给出的前序遍历顺序相反。

5、选 B

在转换规则中，一个结点的右兄弟会成为他的右孩子，所以没有右兄弟的结点在转换完成之后会没有右孩子。

6、选 C

基邻接矩阵反映顶点间邻接关系，设 G=(V，E)是具有 n(n≥1)个顶点的图，C 的邻接矩阵 M 是一个 n 行 n 列的矩阵，并有若(i，j)或<i，j>∈E，则 M[i][j]=1：否则， M[i][j]=0。

由邻接矩阵的定义可知无向图的邻接矩阵是对称的，即图中的一条边对应邻接矩阵中的两个非零元素。因此，在一个含有 n 个顶点和 e 条边的简单无向图的邻接矩阵中共有 n^2-2e 个零元素。

7、选 B

一棵高度为 2 的 7 阶 B 树，根结点只有到达 7 个关键字的时候才能产生分裂，成为高度为 2 的 B 树。

8、选 C

基本概念

9、选 B

本题考查 B 树和 B+树的概念和特点。

B 树的定义是这样的，一棵 m 阶的 B 树满足下列条件：

(1)每个结点至多有 m 棵子树；

(2)除根结点外，其他每个非叶子结点至少有 m/2 棵子树；

(3)若根结点不是叶子结点，则至少有两棵子树；

(4)所有叶结点在同一层上。B 树的叶结点可以看成一种外部结点，不包含任何信息；

(5)所有的非叶子结点中包含的信息数据为：$(n,p_0,k_1,p_1,k_2,P_2,\cdots, k_{j-1},P_{j-1})$

其中，k_i 为关键字，且满足 $k_i<k_{i+1}$;p_i 为指向子树根结点的指针，并且 P_{i-1} 所指的子树中的所有结点的关键字均小于 k_i，P_{j-1} 所指的子树中的所有结点的关键字均大于 k_{j-1}。

B+树是应文件系统所需而出现的一种 B 树的变型树，其主要区别是一棵非叶子结点有 n 个子树就有 n 个关键字，这些关键字的作用是索引；所有的叶子结点包含了全部关键字的信息，以及指向这些关键字记录的指针，且叶子结点本身的关键字的大小自小而大顺序链接。

从上述的特点中我们知道，这两种树都是平衡的多分树，它们都可以用于文件的索引结构，但 B 树只能支持随机检索，而 B+树是有序的树，既能支持随机检索，又能支持顺序检索。

10、选 C

因为概率都相同所以最后的构建就是一颗满二叉树。

二、填空题(每空 1 分,共 10 分)

答案概览:

11__n+1__　　12__线性查找和折半查找、$\sqrt{n}$__

13__$\log n$ 、$n\log n$__

14__O（m+e）__　　15__极大连通子图__

16__5__　　17__从源点到汇点路径长度最长的路径。__

详细解答:

11、

对于 n 个结点的二叉树一共有 2*n 个结点，我们知道连接 n 个点只需要 n-1 条边，也就是利用 n-1 个结点进行连接，剩下 n+1 个节点即位空子树。

12、13

均为基本概念，对于 k 路归并排序对 n 个元素，需要进行$\log_k n$趟。

14、基本概念。

15、

生成树指的是含有所有顶点的无环连通子图。注意这其中的三个限定条件：

1）包含了所有的顶点

2）不存在环

3）连通图

16

有 n 个结点的AVL树的高度不超过：

$$h = \log_{\varphi}(\sqrt{5}(n+1)) - 2 \qquad \varphi = \frac{1+\sqrt{5}}{2}$$

$$\approx 1.44 * \log_2 (n+1)$$

最多的比较次数就是树的高度，我们把 12 带入公式，这个公式大家记住就可以了

17、记忆类题目

三、简单题（每小题 10 分，共 20 分）

详细解答:

20.

堆排序：取 n 个数中的前 k 个建立小根堆，然后在剩下的元素中依次与堆顶元素比较，若比堆顶小则不可能在前 k 个最大值中，直接丢弃。若比对顶大则替换掉堆顶，并重新将其调整为小根堆。直到剩下的元素全部比较完，堆中的 k 个数即为最大。

T（n）=O(nlogk),S(n)=k

败者树：建立一个叶子结点个数为 k 的败者树，较小者为胜者，则最顶层保存的是值最小的叶子结点，剩下的元素依次与最小的叶子结点比较，若比 min 小则直接丢弃，否则用该元素替换树中的最小值，并从上往下调整败者树，直到全比较完毕，树中剩下的 K 个叶子结点即为 k 个最大值

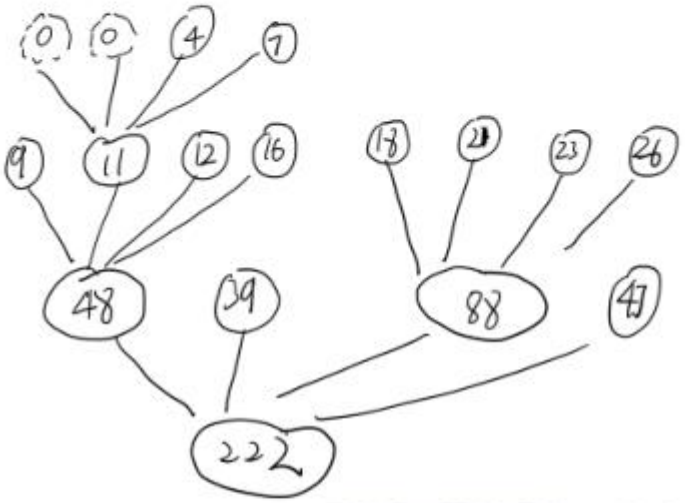
T(n)=O(nlogk),S（n）=O（k）

21.

1）英文名 26 个字母设置 26 个队列，根据名字将发票和资料分别进行基数排序，排好序后将发票和资料意义对应分为各组即可。

2）根据英文名各位字母基数排序所得的次序给参会者编号，并将分好组的待发资料和发票也编号，按序号发给对应人员

22. k=4，m=11，k-(m-1) mod (k-1)-1=2



四、算法设计题（23 题 13 分，24 题 12 分，共 25 分）

23 题

具体的数据结构如下

```
struct stack
{
    int top;
    ElemType data[max_size];
}stk;
struct stack
{
    int top;
    ElemType data[max_size];
}max_stk;
```

（1）实现 Push 操作

```
void push(ElemType v)
{
    stk.data[stk.top++]=v;
    if(max_stk.top==0)max_stk.data[max_stk.top++]=v;
    else
    {
        ElemType tmp=max_stk.data[max_stk.top];
        if(tmp<v)
        {
            max_stk.data[max_stk.top]=v;
        }
        else
        {
            max_stk.data[max_stk.top]=tmp;
        }
    }
}
```

（2）实现 Pop 操作

```
void pop()
{
    if(stk.top!=0)
    {
        ElemType=stk.data[stk.top--];
    }
    if(max_stk.top!=0)
    {
        max_stk.data[max_stk[max_stk.top-1]];
    }
}
```

（3）获取最大值

```
ElemType getMax()
{
    return max_stk.data[max_stk.top];
}
```

24 题

```
typedef struct BTNode{
    int data;
    struct BTNode *lchild,*rchild;
}BTNode,*BiTree;

int findNearMid(BiTree root){
    int half,min,max;BTNode *p,*q=root;
    while(p!=NULL) p=p->lchild;
    while(q!=NULL) q=q->rchild;
    min= p->data;max=q->data;half=(max+min)/2;
    while(T!=NULL){
        if(T->data<=half) T=T->rchild;
        else{
p=T;
if(T->lchild==NULL) return p->data;
else T=T->lchild;
        }
    }
    return p->data;
}
```