

# 14-15 学年第 2 学期期末《数据结构》试题 A 参考答案

## 一、单项选择 (15 × 2 = 30分)

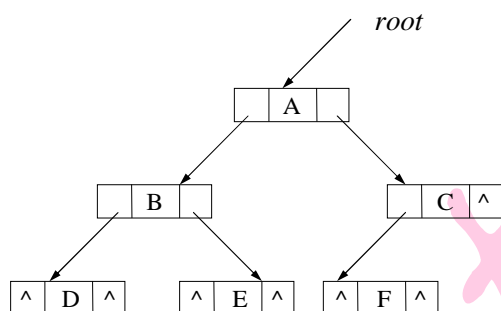
D C C C B      A D A B C      D D A D C

## 二、填空 (10 × 2 = 20分)

- 1、存储(或者 物理)结构    2、 $O(1)$     3、 $n-i+1$     4、186    5、叶子  
6、 $2m-1$     7、邻接(多重)表    8、链地址法    9、 $\frac{n+1}{2}$     10、归并排序

## 三、解答题 (5 × 10 = 50分)

1、



先序: ABDECF、中序: DBE AFC、后序: DEBFCA

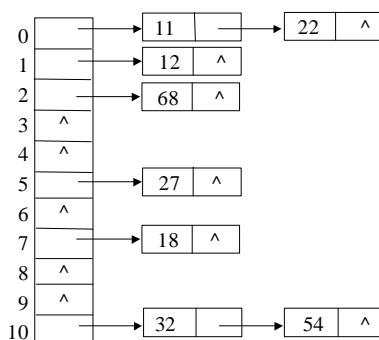
2、(1)

下标	0	1	2	3	4	5	6	7	8	9	10
关键字	11	12	22	54	68	27		7			32
成功比较次数	1	1	3	5	3	1		1			1
失败探测空位次数	7	6	5	4	3	2	1	2	1	1	8

$$ASL_{succ} = \frac{1}{8} (5 \times 1 + 2 \times 3 + 1 \times 5) = 2$$

$$ASL_{unsucc} = \frac{1}{11} (1 + 2 + \dots + 8 + 1 + 2 + 1) = \frac{40}{11} \approx 3.64$$

(2)



$$ASL_{succ} = \frac{1}{8} (6 \times 1 + 2 \times 2) = \frac{10}{8} = 1.25$$

$$ASL_{unsucc} = \frac{1}{11} (5 \times 0 + 4 \times 1 + 2 \times 2) = \frac{8}{11} \approx 0.727$$

3、从顶点 D 出发的深度优先搜索遍历序列为: DECBA

(2 分)

从顶点 B 出发的广度优先搜索遍历序列为: BCAED

(2 分)

- 4、(1) 第一趟快速排序结果为：20, 17, 12, 5, 22, 35, 41, 30 (2 分)  
(2) 初始大顶堆为：41, 35, 30, 20, 22, 5, 12, 17 (2 分)  
(3) 第一趟归并排序结果为：17, 22, 30, 41, 5, 35, 12, 20  
第二趟归并排序结果为：17, 22, 30, 41, 5, 12, 20, 35  
第三趟归并排序结果为：5, 12, 17, 20, 22, 30, 35, 41 (4 分)

四、算法设计 (2 × 10 = 20分)

1、

```
void Merge(LinkList ha, LinkList hb, LinkList *hc)
{
    Node *s = NULL;
    *hc = NULL;
    while (ha != 0 && hb != 0)
    {
        if (ha->data < hb->data)
        {
            if (s == NULL)
            {
                *hc = ha; s = ha;
            }
            else
            {
                s->next = ha; s = ha;
            }
            ha = ha->next;
        }
        else
        {
            if (s == NULL)
            {
                *hc = hb; s = hb;
            }
            else
            {
                s->next = hb; s = hb;
            }
            hb = hb->next;
        }
    }
    s->next = (ha == NULL ? hb : ha);
}
```

2、

```
int Depth(BTree root)
{
    int depth = 0, left, right;
    if (root)
    {
        left = Depth(root->lchild);
        right = Depth(root->rchild);
        depth = 1 + (left > right ? left : right);
    }
    return depth;
}
```