《数据结构 B 类（A 卷）》参考答案

一. 选择题：

1. A　2. B　3. D　4. C　5. A　6. A　7. B　8. B　9. C　10. A

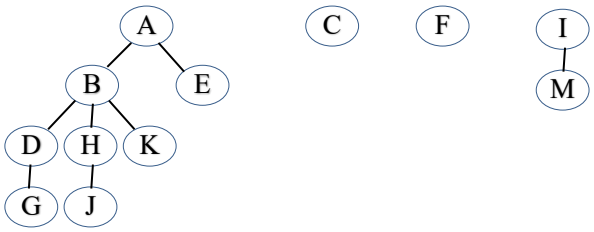
二. 填充题：

1. O(1)　2. 6　3. 5+(1+2)*4-3　4. 7　5. 2^{k-1}、2^k-1　6. PPDPDDPDPD

7. 7, 4, 2, 5, 16, 18, 20, 29, 33, 40, 34　8. 直接插入或冒泡算法　9. (n +1)/2

三. 简答题：

1. 第一棵树 2 分，之后 3 棵树各 1 分



2. **功能（3 分）**：在图 G 中从结点 i 出发的深度优先遍历；**复杂度（2 分）**：$O(N^2)$

四. 分析题：

1.

1) 返回 1；

2) 此函数是判断链表 hb 是否是链表 ha 前缀的函数。若 hb 是链表 ha 前缀将返回数值 0，否则返回的函数值为 1；

3) O(min(LinkList a.length, LinkList b.length)　**（注意：此小题如果写成 O(n)，得 2 分）**

2.

1) 函数 B 是删除以 t 为根的树**（3 分，一定要说明删除的是“以 t 为根的树”，否则扣 1 分）**

2) 函数 A 删除当前 BST 的其他结点，保留以 x 为根结点的部分**（3 分，如果没有说明保留的是“以 x 为根结点”，则扣 1 分）**

五. 程序填充题：

1. 1) Insert(x, root)；2) insert(x, t->left)；3) insert(t->right)

2. 1) p = p->left；2)p= p->right；3) p != NULL

3. 1) current->LeftChild；2) number += childnumber(current->LeftChild) + 1

3) current->RightChild；4) number += childnumber(current->RightChild) + 1

六. 编程题：

1.

1) 由于只知道链表头指针，因此只能顺序遍历链表。但需要访问倒数第 k 个位置上结点，即“先入后出”，因此可采用栈作为辅助存储结构，实现步骤如下：

a) 遍历链表，将结点指针进栈，并统计链表中结点总个数 total。

b) 如果 total < k, 即不存在倒数第 k 个位置上的结点，返回 0。

c) 出栈 k-1 次，栈项位置结点即为所求，输出其 data 域，并返回 1。

注意：此小题考查“链表+栈”的应用，如果不用栈，但算法正确的，则扣 1 分。如果算法不正确的话，不得分。

2) C++语言描述如下：

```
#include <iostream>
#include <stack>
using namespace std;
int LocateElement (Linklist& list,int k) {
    stack <LNode*> S;
    LNode* p, *q;
    p = list->link;
    int count = 0;
    while (p != NULL) {                    //进栈，并统计链表中元素个数
        count++;
        S.push (p);
        p = p->link;
    }
    if (count < k) return 0;               //链表中结点总数小于k
    while (--k) S.pop ();                  //出栈k - 1次
    cout << S.top->data << endl;           //打印倒数第k个元素数据
    return 1;
}
```

注意：如果不用栈实现，但程序正确的话，则扣 2 分。如果程序不正确，得分不能超过 3 分。

3) 算法的时间复杂度 O(n)。

注意：如果没有使用“链表+栈”的，且算法正确的话，至多得 2 分。

2. 利用在每趟排序中进行正向和反向两遍冒泡的方法一次可以得到两个最终值(最大者和最小者)

```
void Bi-BubbleSort (int a[], int size) {
    int count = 0;
    int low = 0;
    int high = size - 1;
    while (low < high) {
        for (int i = low; i < high; i++) {
            if (a[i] > a[i + 1]) {
                temp = a[i];
                a[i] = a[i + 1];
                a[i + 1] = temp;
            }
        }
        high--;
        for (int j = high; j > low; j--) {
```

```
            if (a[j] < a[j - 1]) {
                temp = a[j];
                a[j] = a[j - 1];
                a[j - 1] = temp;
            }
        }
        low++;
        count++;
    }
    for (int i = 0; i < size; i++) {cout << a[i] << " ";}
    cout << endl;
}
```

注意：

1) 程序一定要体现正向和方向同时排序，否则即使程序达到排序功能，但最多得 5 分。

2) 如果只写思路或算法，且思路或算法是正确的话，则最多得 2 分。

3) 如果程序不正确，则最多得 2 分。

4) 如果算法正确，程序不正确，则最多得 4 分。