

Homework 12.7 12.9

PB19010450 和泳毅

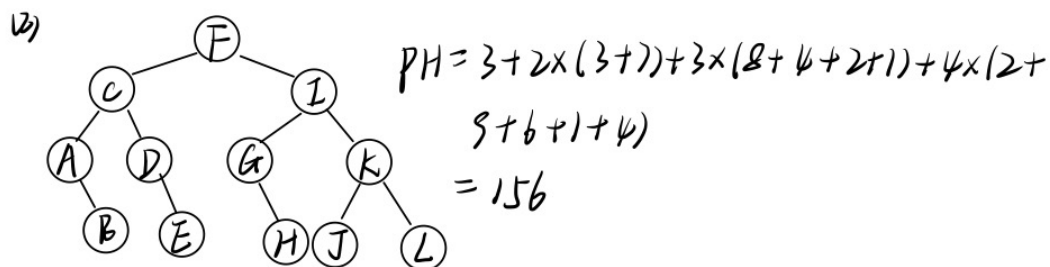
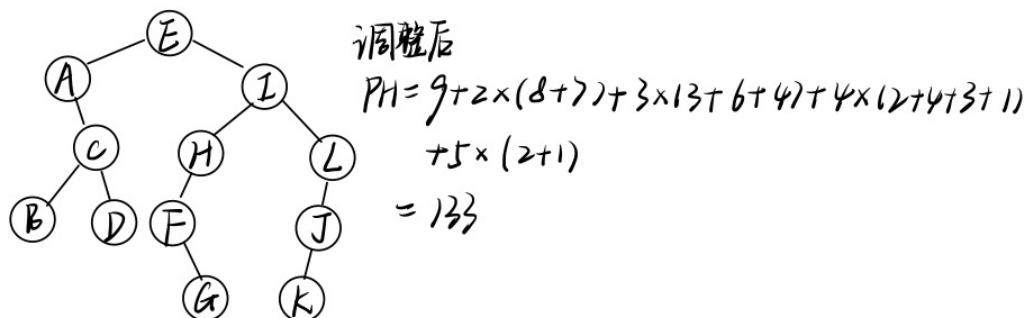
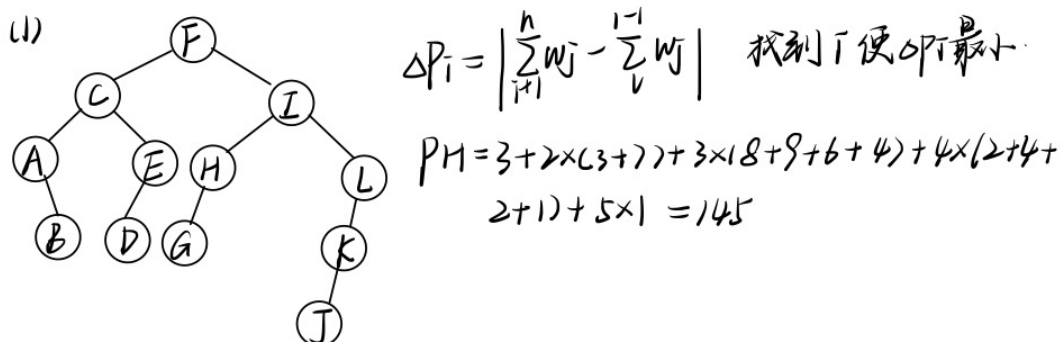
◆ 9.8③ 已知含 12 个关键字的有序表及其相应权值为：

关键字	A	B	C	D	E	F	G	H	I	J	K	L
权值	8	2	3	4	9	3	2	6	7	1	1	4

(1) 试按次优查找树的构造算法并加适当调整画出由这 12 个关键字构造所得的次优查找树，并计算它的 PH 值；

(2) 画出对以上有序表进行折半查找的判定树，并计算它的 PH 值。

(1)



```

1  int bin = 0; //0表示没有找到
2  Status Search_Bin(SSTable T, keyType key, int low, int high){
3      //low初始1, high初始ST.length
4      if(low > high) return FALSE;
5      mid = (low + high) / 2;
6      if(T.elem[mid].key == key) bin = mid;
7      else if(T.elem[mid].key > key) Search_Bin(T, key, low, mid - 1);
8      else Search_Bin(T, key, mid + 1, high);
9      return OK;
10 }

```

9.31 试写一个判别给定二叉树是否为**二叉排序树**的算法，设此二叉树以**二叉链表**作存储结构。且树中结点的关键字均不同。

```

1  int flag = TRUE;
2  Status IS_BST(BiTree T){
3      if(!T) return TRUE; //空树也是BST
4      if(T->lchild && flag){
5          if(T->lchild->data.key > T->data.key) flag = FALSE;
6          else IS_BST(T->lchild);
7      }
8      if(T->rchild && flag){
9          if(T->rchild->data.key < T->data.key) flag = FALSE;
10         else IS_BST(T->rchild);
11     }
12     return OK;
13 }

```

9.33 编写递归算法，从大到小输出给定二叉排序树中所有关键字**不小于x**的数据元素。要求你的算法的时间复杂度为 $O(\log_2 n + m)$ ，其中n为排序树中所含结点数，m为输出的关键字个数。

```

1  void NLS_x(BSTree T, int x){
2      if(T){
3          NLS_x(T->rchild, x); //从大到小
4          if(T->data.key >= x) printf("%d ", T->data.key);
5          else return; //比x小即可返回,不用递归此结点的左子树
6          NLS_x(T->lchild, x);
7      }
8  }

```

9.34 试写一时间复杂度为 $O(\log_2 n + m)$ 的算法，删除二叉排序树中所有关键字**不小于x**的结点，并释放结点空间。其中n为树中所含结点数，m为被删除的结点个数。

```

1  void Delete_NLT_x(BSTree T, int x){
2      if(T){
3          if(T->data.key >= x){
4              p = T;
5              T = T->lchild;
6              p->lchild = NULL;
7              Free_NLT_x(p);
8              Delete_NLT_x(T->lchild, x);
9          }

```

```
10         else Delete_NLT_x(T->rchild,x);
11     }
12 }
13 void Free_NLT_x(BSTree T){
14     if(T){
15         Free_NLT_x(T->lchild);
16         Free_NLT_x(T->rchild);
17         free(T);
18         T = NULL;
19     }
20 }
```
