DS Homework 13

注: 请使用 A4 纸作答,写上姓名学号,并于下一次上课时提交。

1、已知无向图 G,给出求距 V0 的各定点中最短路径长度最长的一个顶点的算法。(伪代码或算法流程皆可)

```
解: 同上例。引入向量空间 len[0..n-1]。其中 len[i]为顶点 v<sub>i</sub> 到 v<sub>0</sub> 的最短路径。
A1. 初始化一个空队列 Q;
A2. 初始化一个向量空间 len[0..n-1]的各个分量初值为∞;
A3. [EnQueue(Q, v<sub>0</sub>); len[0]←0; 设置 v<sub>0</sub> 为已访问标志;]
A4. 循环: 当队列 Q 不为空时,重复执行
    [A4.1. DeQueue(Q, v<sub>i</sub>);
    A4.2. 循环: 当 v<sub>i</sub> 存在未访问的邻接点 v<sub>j</sub> 时,重复执行
    [A4.2.1. EnQueue(Q, v<sub>j</sub>);
    A4.2.2. 设置 v<sub>j</sub> 为已访问标志;
    A4.2.3. len[v<sub>j</sub>]←len[v<sub>i</sub>]+1;
]

A5. [m←len[0]; pos←0; ]
A6. 循环: i以 1 为步长,从 1 到 n-1,重复执行
    [如果 m<len[i],则 m←len[i]; pos←i;]
    [如果 m<len[i],则 m←len[i]; pos←i;]
A7. 输出距 v<sub>0</sub> 的各顶点中最短路径长度中最长的一个顶点 pos;
A7. 输出距 v<sub>0</sub> 的各顶点中最短路径长度中最长的一个顶点 pos;
A8. 算法结束.
```

2、试写一个算法,判定给定的关键字序列 k1, k2,..., kn 是否为折半查 找过程中可能出现的关键字比较序列。(伪代码或算法流程皆可) 解: 将给定的关键字序列 k_1 , k_2 , …, k_n 放入 v[1..n]中。假设折半查找是在有序表 list[low..high] 对应的折半表 list[low..high] 上进行的。与本题等价的描述为在有序表 list[low..high] 对应的折半查找树上是否存在着从根结点开始,到某一个结点的路径上每个结点的关键字序列与查找树上是否存在着从根结点开始,到某一个结点的路径上每个结点的关键字序列与 list k_1 , k_2 , …, k_n 相同。判定过程为:首先用 v[1]与 list[L(low+high)/2]] 不相等则就可 [L(low+high)/2]] 为折半查找树的根。如果 v[1]与 list[L(low+high)/2]] 不相等则就可 v[1] 为 v[2] 为 v[2] 与 v[2]

判断关键字序列 k_1 , k_2 , …, k_n 是否为在有序表 list[low..high]上进行折半查找可能出现的关键字比较序列。如果是输出"是", 否则输出"不是"。算法描述如下:

```
A1. v[1..n] ← (k<sub>1</sub>, k<sub>2</sub>, ···, k<sub>n</sub>);
A2. result ← 1; i ← 1; a ← low; b ← high;
A3. 循环: 当 result = 1 且 i ≤ n 时, 重复执行
[A3.1. 如果 v[i] ≠ list[⌊(a+b)/2⌋], 则 [ result ← 0; 转(A3.4); ]
A3.2. 如果 i < n, 则
[如果 v[i+1] > list[⌊(a+b)/2⌋], 则 a ← ⌊(a+b)/2⌋;
否则 b ← ⌊(a+b)/2⌋;
]
A3.3. i ← i + 1;
A3.4. 空操作;
```

A4. 如果 result = 1,则 输出"是",否则 输出"不是";

A5. 算法结束.