

数据结构 技术报告内容

一. (共 32 分) 二叉搜索树或二叉排序树(以下简称 BST)是一种动态数据结构, 请回答下面的问题(假设 BST 中没有重复元素)。

1. (8 分) 对于集合{1, 4, 5, 10, 16, 17, 21}, 请画出树高分别满足为 3,4,5,6 的一棵 BST。

2. 现给定如下树结点结构

```
struct Node {  
    Node * left;    //指向结点的左儿子  
    Node * right;   //指向结点的右儿子  
    Node * p;       //指向结点的父亲结点  
    int key;  
};
```

2.1 (4 分) 试分析 BST 中最小元素的位置, 并补全下列伪代码: 该函数返回以结点 x 为根的一棵 BST 中具有最小元素的结点。

```
TREE_MINIMUM(x)  
1   while (_____  
2.   _____  
3.   return x
```

2.2 (4 分) 请写出下列函数的作用, 并分析其时间复杂度, 其中 x 为 BST 中的某个结点。

```
FUNCTION (x)  
1.   if x.right != NIL  
2.       return TREE_MINIMUM(x.right)  
3.   y = x.p  
4.   while y!= NIL and x == y.right  
5.       x = y  
6.       y = y.p  
7.   return y
```

2.3 (4 分) 以下代码将向二叉树 T 中插入一个 key 值为 v 的新结点 z (z.key = v), 请分析它的时间复杂度和空间复杂度(要有分析过程, 不能只写结论)。

```

TREE-INSERT(T, z)
1  y = NIL
2  x = T.root
3  while x ≠ NIL
4      y = x
5      if z.key < x.key
6          x = x.left
7      else x = x.right
8  z.p = y
9  if y == NIL
10     T.root = z      // tree T was empty
11  elseif z.key < y.key
12     y.left = z
13  else y.right = z

```

2.4 (12 分) 以下代码为 2.3 中函数的递归版实现, 请给出递归核心部分 TREE_INSERT_R 函数的伪代码, 并分析其时间复杂度和空间复杂度(要有分析过程, 不能只写结论)。

```

TREE_INSERT(T, z)
1.  if T.root == NIL
2.      T.root = z
3.      z.p = NIL
4.  else TREE_INSERT_R(T.root, z)

```

二. (共 36 分) 下面函数给出了对于一个字符集 C 的 Huffman 编码过程。其中 C 为含有 n 个字符的集合。函数返回一棵 Huffman 树的根结点, 树中每个结点的属性 *left*, *right*, *freq* 分别表示左儿子, 右儿子以及该结点的频率。

```

HUFFMAN(C)
1  n = |C|
2  Q = C
3  for i = 1 to n - 1
4      allocate a new node z
5      z.left = x = EXTRACT-MIN(Q)
6      z.right = y = EXTRACT-MIN(Q)
7      z.freq = x.freq + y.freq
8      INSERT(Q, z)
9  return EXTRACT-MIN(Q)    // return the root of the tree

```

上述函数使用了某种数据结构 Q , 它提供两个函数: EXTRACT-MIN(Q) 返回结构中具有最小 *freq* 的结点, INSERT(Q, z) 则将结点 z 加入到该结构中。

1. (14 分) 如果 Q 是用双向链表实现的, 请给出 EXTRACT-MIN 实现的伪代码, 并分析 HUFFMAN(C) 函数的时间复杂度(要有分析过程, 不能只写结论)。

2. Q 还可以采用基于最小堆(Binary Min Heap)的优先队列来实现。最小堆的逻辑结构是一棵完全二叉树, 其物理结构通常为一个数组。

现规定对于一个含有 n 个元素的最小堆 A , 其元素的下标为 $1, \dots, n$ 。

2.1 （3 分）请写出求第 i 个元素的左儿子，右儿子以及父亲的下标的函数 $\text{Left}(i)$, $\text{Right}(i)$, $\text{Parent}(i)$.

2.2 （9 分）下列 $\text{Build-Min-Heap}(A)$ 将一个数组 A 调整为最小堆，其中用到的 $\text{Min-Heapify}(A,i)$ 将以 i 为根的二叉树做必要的调整使其满足最小堆的属性，该函数假设以 $\text{Left}(i)$ 和 $\text{Right}(i)$ 为根的二叉树已经是最小堆， $A.\text{length}$ 为数组 A 的长度。请写出 $\text{Min-Heapify}(A, i)$ 的伪代码，并分析 $\text{Build-Min-Heap}(A)$ 的时间复杂度(要有分析过程，不能只写结论)。(注意数组为 1-based，即第一个元素为 $A[1]$ ；可以直接使用 2.1 中的 Left , Right 函数。)

```
Build-Min-Heap(A)
1.  for  $i = A.\text{length} / 2$  downto 1
2.  Min-Heapify(A,  $i$ )
```

2.3 （6 分） $\text{Extract-Min}(A)$ 函数完成从最小堆 A 中删除最小元素的操作，请写出其伪代码；

$\text{Insert}(A,z)$ 函数向一个最小堆 A 中增加一个元素 z ，得到的新数组依然保持最小堆的性质，请写出其伪代码。可以使用 2.1，2.2 中出现的函数。

2.4 （4 分）在 Q 为最小堆优先队列的情况下，请分析 Huffman 过程的时间复杂度(要有分析过程，不能只写结论)。

三.（共 32 分）国家为了振兴因疫情而受到影响的经济，制定了新基建计划。该计划希望全国任何两个省会之间都可以实现高铁通达（不一定有直接的高铁相连，只要能间接通过高铁到达即可）。经过调查评估，得到的统计表中列出了有可能建设高铁的若干条路线的成本。现请你编写程序，计算出全国畅通需要的最低成本。

测试用例格式：

第 1 行给出评估的线路条数 N 、省会城市数目 $M (< 100)$ ；

随后的 N 行对应省市间路线的成本，每行给出一对正整数，分别是两个城市的编号以及此间道路的成本（也是正整数）。为简单起见，城市从 1 到 M 编号。

程序将输出全国畅通需要的最低成本。若统计数据不足以保证畅通，则输出“?”。

比如对于用例

1 3
2 3 2

程序将输出？

对于

3 3
1 2 1
1 3 2
2 3 4

程序将输出 3

要求： 请用你所学过的数据结构相关知识来表达该实际问题；给出解决该问题的一种方法(包括所用的数据结构以及相应的算法)， 并分析其复杂度。编写测试用例来验证你程序的正确性。