

实 验 报 告

(与程序设计有关)

课程名称： 数据结构

实验题目： 树

班级学号： 2203050320

姓 名： 闻家尉

成 绩：

沈 阳 理 工 大 学

2023 年 11 月 13 日

实验目的及要求：

1. 熟悉 Huffman 编码方法；
2. 了解并弄懂 Huffman 编码实现信息的无损压缩原理。

软硬件环境： window 11, sublime text 4

算法或原理分析（实验内容）：

1. 问题描述

根据给定的 n 个权值构造哈夫曼树。通过遍历此二叉树完成哈夫曼编码。

2. 基本要求

输入 n 个结点的权值，构造哈夫曼树，输出哈夫曼编码。

3. 算法提示

①根据给定的 n 个权值 (w_1, w_2, \dots, w_n) 构成 n 棵二叉树的集合 $F = \{T_1, T_2, \dots, T_n\}$ ，其中每棵二叉树 T_i 中只有一个带树为 T_i 的根结点；

②在 F 中选取两棵根结点的权值最小的树作为左右子树构造一棵新的二叉树，且置其根结点的权值为其左右子树权值之和；

③在 F 中删除这两棵树，同时将新得到的二叉树加入 F 中；

④重复②、③直到 F 只含一棵树为止。

程序代码或实现过程:

```
#include<bits/stdc++.h>
using namespace std;
int n;
typedef struct
{
    int weight;
    int parent, lchild, rchild;
}HTNode, *HuffmanTree;

typedef char **HuffmanCode;
void Select(HuffmanTree HT, int len, int &s1, int &s2)
{
    int i, min1 = 0x3f3f3f3f, min2 = 0x3f3f3f3f;
    for (i = 1; i <= len; i++)
    {
        if (HT[i].weight < min1 && HT[i].parent == 0)
        {
            min1 = HT[i].weight;
            s1 = i;
        }
    }
    int temp = HT[s1].weight;
    HT[s1].weight = 0x3f3f3f3f;
    for (i = 1; i <= len; i++)
    {
        if (HT[i].weight < min2 && HT[i].parent == 0)
        {
            min2 = HT[i].weight;
            s2 = i;
        }
    }
    HT[s1].weight = temp;
}
void CreatHuffmanTree(HuffmanTree &HT, int n)
{
    int m, s1, s2, i;
    if (n <= 1)
        return;
    m = 2 * n - 1;
    HT = new HTNode[m + 1];
    for (i = 1; i <= m; ++i)
    {
```

```

        HT[i].parent = 0;  HT[i].lchild = 0;  HT[i].rchild = 0;
    }
    cout << "请输入叶子结点的权值: \n";
    for (i = 1; i <= n; ++i)
        cin >> HT[i].weight;
    for (i = n + 1; i <= m; ++i)
    {
        Select(HT, i - 1, s1, s2);
        HT[s1].parent = i;
        HT[s2].parent = i;
        HT[i].lchild = s1;
        HT[i].rchild = s2;
        HT[i].weight = HT[s1].weight + HT[s2].weight;
    }
}

void CreatHuffmanCode(HuffmanTree HT, HuffmanCode &HC, int n)
{
    int i, start, c, f;
    HC = new char*[n + 1];
    char *cd = new char[n];
    cd[n - 1] = '\0';

    for (i = 1; i <= n; ++i)
    {
        start = n - 1;
        c = i;
        f = HT[i].parent;
        while (f != 0)
        {
            --start;
            if (HT[f].lchild == c)
                cd[start] = '0';
            else
                cd[start] = '1';
            c = f;
            f = HT[f].parent;
        }

        HC[i] = new char[n - start];
        strcpy(HC[i], &cd[start]);
    }

    delete cd;
}

```

```
void show(HuffmanTree HT, HuffmanCode HC)
{
    for (int i = 1; i <= n ; i++)
        cout << HT[i].weight << "编码为" << HC[i] << endl;
}

int main()
{
    HuffmanTree HT;
    HuffmanCode HC;

    cout << "请输入叶子结点的个数: \n";
    cin >> n;
    CreatHuffmanTree(HT, n);
    CreatHuffmanCode(HT, HC, n);
    show(HT, HC);

    system("pause");

    return 0;
}
```

结果分析：

```
请输入叶子结点的个数：
5
请输入叶子结点的权值：
1 2 3 4 5
1编码为010
2编码为011
3编码为00
4编码为10
5编码为11
请按任意键继续. . .
```

教师签字	杨书彤	日期	
------	-----	----	--