

Huffman Encoding

采用优先级队列实现哈弗曼树

```
#include <iostream>
#include <queue>
#include <string.h>
#include <vector>
#include <algorithm>

using namespace std;

char Table[26];

struct Node {
    int freq;
    char val;
    Node *left;
    Node *right;

    Node() : left(NULL), right(NULL), freq(0), val('0') {}
};

class Cmp {
public:
    bool operator()(const Node *a, const Node *b) const {
        return a->freq > b->freq; // 反向定义
    }
};

priority_queue<Node *, vector<Node *>, Cmp> myQueue;

void BuildTree(string& keys, vector<int>& freqs) {
    for (int i = 0; i < keys.size(); ++i) {
        Node *node = new Node();
        node->freq = freqs[i];
        node->val = keys[i];
        myQueue.push(node);
    }

    while (myQueue.size() > 1) {
        Node *f = myQueue.top();
        myQueue.pop();
        Node *s = myQueue.top();
        myQueue.pop();
        Node *tmp = new Node();
        tmp->freq = f->freq + s->freq;
        tmp->left = f;
        tmp->right = s;
        myQueue.push(tmp);
    }
}
```

```

struct PrintNode {
    int freq;
    char val;
    string code;
};

// 前序遍历输出哈弗曼树
void printPreOrder(Node* root, string s, int& count) {
    if (root == NULL)
        return;
    if (root->left) {
        s += '0';
        printPreOrder(root->left, s, count);
    }
    if (root->left == NULL && root->right == NULL) {
        if (count == 0) {
            printf("%s", s.c_str());
            count++;
        } else {
            printf(" %s", s.c_str());
        }
    }
    s.erase(s.end() - 1);
    if (root->right) {
        s += '1';
        printPreOrder(root->right, s, count);
    }
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int freq;
        string keys;
        cin >> keys;
        vector<int> freqs;
        for (int i = 0; i < keys.size(); ++i) {
            scanf("%d", &freq);
            freqs.push_back(freq);
        }
        BuildTree(keys, freqs);
        Node* root = myQueue.top();
        myQueue.pop();
        string str = "";
        int count = 0;
        printPreOrder(root, str, count);
        printf("\n");
    }
    return 0;
}

```

关于优先级队列：

浅谈C++ STL中的优先队列(priority_queue) - CieloSun - 博客园

<https://www.cnblogs.com/cielosun/p/5654595.html>