

数据结构作业(第十四次)

PB20111686 黄瑞轩

10.30

先写一个分区函数

```
int partition(vector<int>& L, int low, int high) {
    int l = low, h = high;
    int tmp = L[l];
    int p = L[l];
    while (l < h) {
        while (l < h && L[h] >= p) h--;
        L[l] = L[h];
        while (l < h && L[l] <= p) l++;
        L[h] = L[l];
    }
    L[l] = tmp;
    return l;
}
```

(1)

```
void qsort(vector<int>& L, int low, int high) {
    bool tag = true; //标记序列是否无序
    int pivot;//pivot位置
    int l_num, r_num;//左右子序列元素个数
    int l = low, h = high;
    stack<pair<int, int>> stk;
    while (tag) {
        pivot = partition(L, l, h);
        if (pivot == l) {
            tag = false;
        }
        if (tag) {
            l_num = pivot - l;
            r_num = h - pivot;
            if (l_num > 1 || r_num > 1) {
                if (l_num <= r_num) {
                    stk.push(pair<int, int>(pivot + 1, h));
                    h = pivot - 1;
                }
                else {
                    stk.push(pair<int, int>(l, pivot - 1));
                    l = pivot + 1;
                }
            }
        }
    }
    if (!tag || (l_num <= 1 && r_num <= 1)) {
        //当前子序列已有序
        if (stk.size()) {
            l = stk.top().first;
            h = stk.top().second;
            stk.pop();
            tag = true;
        }
    }
}
```

(2)

```
void qsort(vector<int>& L, int low, int high) {
    int pivot;
    int l_num, r_num;
    int l = low, h = high;
    stack<pair<int, int>> stk;
    while (1) {
        if (h - l + 1 <= 3) {
            sort(L, l, h);//普通排序
            if (stk.size()) {
```

```
        l = stk.top().first;
        h = stk.top().second;
        stk.pop();
    }
    else break;
}
else {
    pivot = partition(L, l, h);
    l_num = pivot - l;
    r_num = h - pivot;
    if (l_num > 1 || r_num > 1) {
        if (l_num <= r_num) {
            stk.push(pair<int, int>(pivot + 1, h));
            h = pivot - 1;
        }
        else {
            stk.push(pair<int, int>(l, pivot - 1));
            l = pivot + 1;
        }
    }
}
}
}
```