

有同学提出，为了提高查找算法的效率，他建议在查找时给给个查找对象增设一个访问频度域，一旦查找到该对象时，将其频度域加1，并按其值的大小确定是否向前移动，使得频度域值越大的越处于查找表的前面。你是否认同改同学的思考，如果认可的话，请帮他设计一个方案包括算法实现他的这个思考。

问题分析：

本质上就是实现找值的同时，记录频度并实时调整序列。

在这里我给出两种算法

算法一

直接使用C++中的vector，每个数据对象内包含两部分：数据值和访问频度。在每一次查询后，由于被查询数的访问频度改变，将其与前面的每一项的访问频度比较并做实时调整，可以直接调用swap函数进行交换，十分方便。

代码实现

```

#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

// 定义查找对象的结构体，包括值和访问频度
struct Obj {
    int value;
    int frequency;
};

// 查找算法
int search(vector<Obj>& list, int target) {
    for (int i = 0; i < list.size(); i++) {
        if (list[i].value == target) {
            // 找到目标对象，增加访问频度
            list[i].frequency++;

            // 比较目标对象的频度和前一个对象的频度，如果目标对象的频度更高，则向前移动
            for (int j = i - 1; j >= 0; j--) {
                if (list[j].frequency < list[i].frequency) {
                    swap(list[j], list[i]);
                    i = j;
                }
                else {
                    break;
                }
            }
            return i; // 返回目标对象的位置
        }
    }
    return -1; // 未找到目标对象
}

int main() {
    vector<Obj> list = { {1, 0}, {2, 0}, {3, 0}, {4, 0}, {5, 0} }; // 初始化对象列表
    int target;
    while (true)
    {
        cout<<"请输入你想要查询的数值"<<endl;
        cin >> target;
        if (target == 000)
            break;
        int pos = search(list, target); // 查找目标对象
        if (pos == -1) {
            cout << "未找到目标对象" << endl;
        }
        else {
            cout << "目标对象的位置为: " << pos << endl;
        }
    }
}

```

```
    }  
}  
return 0;  
}
```

结果展示

C:\Users\lenovo\Desktop\Project1\Debug\Project1.exe

请输入你想要查询的数值
5
目标对象的下标为：0
目前序列为：
5 1 2 3 4
请输入你想要查询的数值
2
目标对象的下标为：1
目前序列为：
5 2 1 3 4
请输入你想要查询的数值
2
目标对象的下标为：0
目前序列为：
2 5 1 3 4
请输入你想要查询的数值
-

算法二

- 运用双向链表的方式
- 如果找到，访问频度加一
- 否则就添加新节点到末尾

代码展示

```

#include <stdio.h>
#include <stdlib.h>
#include <iostream>

using namespace std;

// 定义链表节点
struct Node {
    int value;
    int frequency;
    struct Node* prev;
    struct Node* next;
};

struct Node* head = NULL;

void init() {
    head = NULL;
}

void cleanup() {
    struct Node* cur = head;
    while (cur != NULL) {
        struct Node* next = cur->next;
        free(cur);
        cur = next;
    }
}

void increment(int value) {
    struct Node* cur = head;
    // 遍历链表查找节点
    while (cur != NULL && cur->value != value) {
        cur = cur->next;
    }
    if (cur != NULL) {
        // 找到节点, 将其频度加1
        cout << "找到节点! " << endl;
        cur->frequency++;
        // 如果频度比前一个节点高, 就交换位置
        while (cur->prev != NULL && cur->frequency > cur->prev->frequency) {
            struct Node* prev = cur->prev;
            struct Node* next = cur->next;
            prev->next = next;
            if (next != NULL) {
                next->prev = prev;
            }
            cur->prev = prev->prev;
            cur->next = prev;
            prev->prev = cur;
        }
    }
}

```

```

        if (cur->prev != NULL) {
            cur->prev->next = cur;
        }
        else {
            head = cur;
        }
    }
}
else {
    // 没有找到节点, 需要插入新节点到链表末尾
    cout << "没有找到节点, 需要插入新节点到链表末尾" << endl;
    struct Node* node = (Node*)malloc(sizeof(struct Node));
    node->value = value;
    node->frequency = 1;
    node->next = NULL;
    if (head == NULL) {
        head = node;
        node->prev = NULL;
    }
    else {
        struct Node* cur = head;
        while (cur->next != NULL) {
            cur = cur->next;
        }
        cur->next = node;
        node->prev = cur;
    }
}
}


void print_list() {
    struct Node* cur = head;
    printf("List: ");
    while (cur != NULL) {
        printf("(%d,%d) ", cur->value, cur->frequency);
        cur = cur->next;
    }
    printf("\n");
}

int main() {
    init();
    int i;
    while (true)
    {
        cout << "请输入你想要查找的数值: (输入000结束)" << endl;
        cin >> i;
        if (i == 000)
            break;
        increment(i);
        cout << "现在的序列为: " << endl;
    }
}

```

```
        print_list();  
        cout << endl;  
    }  
    cleanup();  
    return 0;  
}
```

结果展示

 C:\Users\lenovo\Desktop\Project1\Debug\Project1.exe

```
请输入你想要查找的数值:  
5  
没有找到节点, 需要插入新节点到链表末尾  
现在的序列为:  
List: (5, 1)  
  
请输入你想要查找的数值:  
4  
没有找到节点, 需要插入新节点到链表末尾  
现在的序列为:  
List: (5, 1) (4, 1)  
  
请输入你想要查找的数值:  
4  
找到节点!  
现在的序列为:  
List: (4, 2) (5, 1)  
  
请输入你想要查找的数值:  
2  
没有找到节点, 需要插入新节点到链表末尾  
现在的序列为:  
List: (4, 2) (5, 1) (2, 1)  
  
请输入你想要查找的数值:
```