中北大学软件学院

**实验报告**

|  |  |
| --- | --- |
| 专 业： | 软件工程 |
| 课程名称： | 数据结构与算法 |
| 班 级： |
| 学 号： |
| 姓 名： |  |
| 辅导教师： | 李华玲 |

2019年03月制

成绩：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实验时间 | 2019年05月 日 时至 时 | 学时数 | 2学时 | |
| 1.实验名称  查找 | | | | |
| 2.实验目的  （1）掌握静态查找表算法（重点掌握折半查找）；  （2）掌握动态查找表——二叉排序树查找算法； | | | | |
| 3.实验内容  **基本内容：**  算法1：采用顺序存储结构创建静态查找表，对查找表进行顺序查找和改进的顺序查找，并对其查找效率进行比较；  算法2：采用顺序存储结构创建静态查找表——有序表，对有序表进行二分查找；  **选作内容：**  编程实现按二叉排序树算法进行查找。 | | | | |
| 4.实验原理  （1）有序表的折半查找过程是：先确定待查记录所在的范围（区间），然后逐步缩小范围直到找到或找不到该记录为止；  （2）二叉排序树查找过程是：首先将给定值和根结点的关键字比较，若相等，则查找成功，否则将依据给定值和根结点的关键字之间的大小关系，分别在左子树或右子树上继续进行查找； | | | | |
| 5.实验源代码  /\*算法1：采用顺序存储结构创建静态查找表，对查找表进行顺序查找和改进的顺序查找，并对其查找效率进行比较；  算法2：采用顺序存储结构创建静态查找表--有序表，对有序表进行二分查找  \*/  #include<stdio.h>  #include<stdlib.h>  #define EQ(a,b) ((a)==(b))  #define LT(a,b) ((a)<(b))  #define LQ(a,b) ((a)<=(b))  typedef int KeyType;  typedef struct{  KeyType key;  }ElemType;  typedef struct{  ElemType \*elem;  int length;  }SSTable;  int create(SSTable \*ST)  {//创建静态表  int i,n;  printf("请输入你要创建的表的长度：\n");  scanf("%d",&n);  ST->elem =(ElemType \*)malloc((n+1)\*sizeof(ElemType));  if(!ST->elem )  return 0;  printf("请输入%d个数据(若采用二分法查找关键字，则顺序输入)。\n",n);  for(i=1;i<=n;i++)  {  scanf("%d",&ST->elem [i].key);  }  ST->length =n;  return 1;  }  int search\_seq(SSTable ST,KeyType key)  {  int i=1;  ElemType \*p;  while(i<=ST.length&&!EQ(ST.elem[i].key,key))  ++i;  if(i<=ST.length)  return i;  else  return 0;  }  int search\_seq2(SSTable ST,KeyType key)  {  int i;  ST.elem[0].key=key;  for(i=ST.length;!EQ(ST.elem[i].key,key);--i);  return i;  }  int search\_bin(SSTable ST,KeyType key)  {  int low=1;  int high=ST.length;  while(low<=high)  {  int mid=(low+high)/2;  if(EQ(key,ST.elem[mid].key))  return mid;  else if(LT(key,ST.elem[mid].key))  high=mid-1;  else low=mid+1;    }  return 0;  }//search\_bin  int main()  {  int i,time;  SSTable ST;  KeyType key;  printf("请输入创建表的类型（0.无序表，1.有序表）：\n");  scanf("%d",&time);  create(&ST);  printf("创建成功！\n");  printf("请输入需查找的关键字：\n");  scanf("%d",&key);  if(time==0)  {  printf("顺序查找关键字结果为：\n");  i=search\_seq(ST,key);  if(i==0)  {  printf("查找失败,表中无此元素\n");  }  else  {  printf("查找成功！\n");  printf("位置是%d\n",i);  }  printf("改进算法顺序查找关键字结果为：\n");  i=search\_seq2(ST,key);  if(i==0)  {  printf("查找失败,表中无此元素\n");  }  else  {  printf("查找成功！\n");  printf("位置是%d\n",i);  }  }  else  {  printf("二分查找关键字结果为：\n");  i=search\_bin(ST,key);  if(i==0)  {  printf("查找失败,表中无此元素\n");  }  else  {  printf("查找成功！\n");  printf("位置是%d\n",i);  }  }  } | | | |
| 6.实验结论及心得  1-顺序表有点遗忘，在调用函数的时候要对应 | | | |