**《数据结构》**

**实验报告6**

|  |  |
| --- | --- |
| **班 级** |  |
|  |  |
| **学 号** |  |
|  |  |
| **姓 名** |  |
|  |  |
| **批阅教师** | **赵宏庆** |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**评分标准：**

**评分成绩为A,B,C,D,E 五档，满分为A**

* + 1. **简答题：每小题回答不完整和错误处超过2条（含2条），总分降一档）**
    2. **填空题：每错两空总分降一档。**
    3. **补交作业成绩降一档**
    4. **抄袭成绩为零，且不准补交**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**实验名称**：查找

**实验地点**：学院机房

**所使用的工具软件及环境**：Dev C++

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

|  |
| --- |
| 1. 实验目的：    1. 通过本实验，加深对查找的理解    2. 掌握查找的存储结构，并掌握顺序查找、二分查找、二叉查找树、哈希表及其查找的算法。 |

|  |
| --- |
| 1. 简答题    1. 顺序查找和折半查找，对初始线性表有什么样的要求。   答：  1. 顺序查找是针对以顺序表或者链式表表示的静态查找表；  2. 折半查找是针对有序表表示的静态查找表，且限于顺序存储结构。   * 1. 折半查找的效率如何？   答：  1. 折半查找是针对有序表进行的查找，假设表中每个记录的查找概率相等，则查找成功的折半查找的平均查找长度为ABLbs=(n+1)/n\*logz(n+1)-1,  2. 当n足够大时，折半查找的平均查找长度为：ABLbs=log2(n+1)-1  3. 折半查找的效率比顺序查找的效率高，但折半查找只适用于有序表，且限于顺序存储结构   * 1. 二叉排序树或者是一棵空树；或者是具有什么样特性的二叉树？   答：  二叉排序树或者是一棵空树，或者是具有下列性质的二叉树：  1. 若它的左子树不空，则左子树所有的节点的值均小于它的根结点的值；  2. 若它的右子树不空，则右子树上所有结点的值均大于它的根结点的值；  3. 它的左右子树也分别为二叉排序树。   * 1. **二叉平衡树**是二叉查找树的另一种形式，其特点是什么？   答：  平衡二叉树或者是一棵空树，或者是具有以下性质的二叉树：它的左子树和右子树都是平衡二叉树，且左子树和右子树的深度之差的绝对值不超过1。 |
| 1. 实验内容   （一）顺序查找算法的实现  对于用户给定的一组关键字序列（64，80，13，56，37，92，19，05，88，21，75），采用顺序查找算法找到给定的关键字，并返回其在查找表中的下标。  #define KeyType int  #define InfoType char  #define SSTABLE\_INIT\_SIZE 100  typedef struct{  KeyType key;  InfoType info;  }ElemType;  typedef struct{  ElemType \*elem;  int length;  }SSTable;  int Search\_Seq(SSTable ST,KeyType kval)  {  int i;  ST.elem[0].key=kval;  for(i=ST.length;i>=0;i--)  {  if(ST.elem[i].key== （1） )  break;  }  return （2） ;  }/\*Search\_Seq\*/  main()  {  SSTable ST;  KeyType kval;  int k;  ST.elem=(ElemType\*)malloc(sizeof(ElemType)\*SSTABLE\_INIT\_SIZE);  ST.elem[0].key=0;  ST.elem[1].key=64;  ST.elem[2].key=80;  ST.elem[3].key=13;  ST.elem[4].key=56;  ST.elem[5].key=37;  ST.elem[6].key=92;  ST.elem[7].key=19;  ST.elem[8].key=05;  ST.elem[9].key=88;  ST.elem[10].key=21;  ST.elem[11].key=75;  ST.length=11;  printf("\n\n");  printf("input the value to be searched:");  scanf("%d",&kval);  k=Search\_Seq( （3） ,kval);  printf("\n\n");  if(k==0)  printf("sorry,not searched!");  else  printf("searched and the position is:%d",k);  }  **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  **答案：**  **（1）kval。**  **（2）i。**  **（3）ST。** | |

|  |
| --- |
| （二）折半查找算法的实现  对于用户给定的一组有序的关键字序列（05，13，19，21，37，56，64，75，80，88，92），采用折半查找算法找到给定的关键字，并返回其在查找表中的下标。  #define KeyType int  #define InfoType char  #define SSTABLE\_INIT\_SIZE 100  typedef struct{  KeyType key;  InfoType info;  }ElemType;  typedef struct{  ElemType \*elem;  int length;  }SSTable;  int Search\_Bin(SSTable ST,KeyType kval)  {  int low,high,mid;  low=1;  high=ST.length;  while(low<=high){  mid=( （1） )/2;  if(kval==ST.elem[mid].key) return mid;  else  if(kval<ST.elem[mid].key)  high=mid-1;  else  low= （2） ;  }/\*while\*/  return 0;  }/\*Search\_bin\*/  main()  {  SSTable ST;  KeyType kval;  int k;  ST.elem=(ElemType\*)malloc(sizeof(ElemType)\*SSTABLE\_INIT\_SIZE);  ST.elem[0].key=0;  ST.elem[1].key=05;  ST.elem[2].key=13;  ST.elem[3].key=19;  ST.elem[4].key=21;  ST.elem[5].key=37;  ST.elem[6].key=56;  ST.elem[7].key=64;  ST.elem[8].key=75;  ST.elem[9].key=80;  ST.elem[10].key=88;  ST.elem[11].key=92;  ST.length=11;  printf("\n\n");  printf("input the value to be searched:");  scanf("%d",&kval);  k=Search\_Bin(ST,kval);  printf("\n\n");  if(k==0)  printf("sorry,not searched!");  else  printf("searched and the position is:%d",k);  }  **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  **答案：**  **（1）low + high。**  **（2）mid + 1。** |

|  |
| --- |
| 1. 教师评语 |