华北电力大学

**实 验 报 告**

|

|

**实验名称** 搜索

**课程名称**  数据结构综合实践

|

|

专业班级： 信息2202 学生姓名：徐梓乔

学 号： 220221100327 成 绩：

|  |
| --- |
| 一、实验内容  给定关键字在无序序列53、78、65、17、87、9、81、45、23、67中进行顺序搜索。  给定关键字在有序序列12、13、21、24、28、30、42、50、69中进行二分搜索。  根据关键字53、78、65、17、87、9、81、45、23、67构造二分搜索树，并给定关键字查找所在的结点，以及对该组关键字进行升序排列。  二、实验方法与步骤  1.顺序搜索:  private void btnSeqSearch\_Click(object sender, EventArgs e)  {  string s = txtSeqSearch.Text.Trim();  try  {  int key = short.Parse(s);  int[] record = new int[10] { 53, 78, 65, 17, 87, 9, 81, 45, 23, 67 };  int num = Search.SeqSearch(record, key);  lblSeqSearch.Text = num.ToString();  }  catch (Exception ex)  {  MessageBox.Show(ex.Message, "错误信息", MessageBoxButtons.OK, MessageBoxIcon.Hand);  txtSeqSearch.Focus();  }  }  public static int SeqSearch<T>(T[] record, T key) where T : IComparable<T>  {  for (int i = 0; i < record.Length; i++)  {  if (record[i].CompareTo(key) == 0)  {  return i;  }  }  return -1;  }  2.二分搜索:  private void btnMidSearch\_Click(object sender, EventArgs e)  {  string s = txtMidSearch.Text.Trim();  try  {  int num = short.Parse(s);  double[] record = new double[9] { 12.0, 13.0, 21.0, 24.0, 28.0, 30.0, 42.0, 50.0, 69.0 };  int num2 = Search.BinSearch(record, num);  lblMidSearch.Text = num2.ToString();  }  catch (Exception ex)  {  MessageBox.Show(ex.Message, "错误信息", MessageBoxButtons.OK, MessageBoxIcon.Hand);  txtMidSearch.Focus();  }  }  public static int BinSearch<T>(T[] record, T key) where T : IComparable<T>  {  int num = 0;  int num2 = record.Length - 1;  while (num <= num2)  {  int num3 = (num + num2) / 2;  if (record[num3].CompareTo(key) < 0)  {  num = num3 + 1;  continue;  }  if (record[num3].CompareTo(key) == 0)  {  return num3;  }  num2 = num3 - 1;  }  return -1;  }  3.二叉树搜索:  private void btnSBT\_Click(object sender, EventArgs e)  {  string s = textBox3.Text.Trim();  try  {  int key = short.Parse(s);  label9.Text = bst[key].ToString();  }  catch (Exception ex)  {  MessageBox.Show(ex.Message, "错误信息", MessageBoxButtons.OK, MessageBoxIcon.Hand);  textBox3.Focus();  }  }  4.中序遍历:  private void button4\_Click(object sender, EventArgs e)  {  int[] array = new int[10] { 3, 2, 1, 4, 5, 6, 7, 10, 9, 8 };  AVLTree<int, int> aVLTree = new AVLTree<int, int>();  for (int i = 0; i < array.Length; i++)  {  aVLTree.Insert(array[i], array[i]);  }  string text = bst.MidOrderTraversal();  label14.Text = text;  }  三、实验结果    四、结论与总结  结论:搜索序列可以使用顺序搜索(序列为无序序列时)、二分搜索(序列为有序序列时)和二叉搜索树.  总结:顺序搜索用于对无序记录的搜索,二分搜索同于对有序记录的搜索.搜索二叉树相较于前两项泛用性较高. |