**安徽大学集成电路学院**

**《算法设计与分析》第二次报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 算法分析与设计 | | | 日期 |  |
| 专业 |  |  |  |  |  |
| 【实验目的】  使用matlab实现二分查找算法，加深对该算法的理解,并评估其性能及适用场景。 | | | | | |
| 【实验原理（预习）】  二分查找是一种非常高效的查找算法,它适用于在有序数组中查找特定元素。  1. 首先将数组中间元素与要查找的元素进行比较:  - 如果中间元素等于要查找的元素,则找到并返回该元素的索引。  - 如果中间元素小于要查找的元素,则在数组的右半部分继续查找。  - 如果中间元素大于要查找的元素,则在数组的左半部分继续查找。  2. 每次将查找范围缩小一半,直到找到目标元素或确定该元素不存在于数组中。  3. 算法的时间复杂度为O(log n),其中n为数组的长度。这是因为在每次迭代中,查找范围都会减少一半,所以需要的迭代次数呈对数增长。  4. 二分查找的前提条件是数组必须是有序的。如果数组未排序,则无法使用二分查找,需要使用其他查找算法,如线性查找。  5. 二分查找通常通过递归或迭代的方式实现。 | | | | | |
| 【实验内容与记录（题号、操作步骤、数据记录与处理、附图编号、代码等）】  **Matlab代码如下**   |  | | --- | | BinarySearch.m | | function index = BinarySearch(T, l, r, x)      while l <= r          m = floor((l + r) / 2);          if T(m) == x              index = m;              return;          elseif T(m) < x              l = m + 1;          else              r = m - 1;          end      end      index = -1;  end |   接下来我们对写好的二分查找算法程序进行测试，观察其是否符合预期  运行结果如下图，结果正确   |  | | --- | | Input.m | | T = [1, 3, 5, 7, 9, 11, 13, 15];  l = 1;  r = length(T);  x = 7;  index = BinarySearch(T, l, r, x) | | | | | | |
| 【小结与讨论】   1. 二分查找算法适用于有序数组,体现出其对数级时间复杂度的优势。在大规模数据处理中,二分查找是首选的搜索算法之一。 2. 该算法需要事先确保数组有序,这增加了预处理的开销。因此,在数组频繁变动的情况下,二分查找可能不太适用。 3. 实验中可以进一步测试边界情况,如数组只有一个元素,或数组为空等情况。 4. 后续可以尝试实现其他查找算法,如哈希查找、树形查找等,并比较它们的性能特点。 | | | | | |