**安徽大学集成电路学院**

**《算法设计与分析》第五次报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 算法设计与分析 | | | 日期 |  |
| 专业 |  |  |  |  |  |
| 【实验目的】  本实验旨在通过使用MATLAB语言实现求两个集合交集的算法，加深对集合运算的理解和掌握，并提升对算法设计与分析的实际应用能力。具体目标包括：  理解集合交集的基本概念：通过编程实现集合交集操作，掌握其数学原理和计算方法。  熟悉MATLAB编程环境：在MATLAB中进行代码编写和调试，增强对MATLAB语言的使用能力。  培养问题解决能力：通过设计和实现算法，提升分析问题、设计算法、编写代码和调试程序的综合能力。  验证算法正确性和效率：通过实验验证算法的正确性，分析算法的时间复杂度和空间复杂度，理解其在实际应用中的性能表现。 | | | | | |
| 【实验原理（预习）】  求两个集合的交集问题，是指找出两个集合中都包含的元素。对于两个集合A和B，它们的交集记为，其定义为包含所有既属于集合A又属于集合B的元素的集合。  数学原理  集合交集的数学定义如下：    算法原理  实现求两个集合交集的算法可以有多种方法，常见的有以下几种：  1.朴素方法：  遍历集合 \( A \) 中的每个元素，对于每个元素，检查它是否存在于集合 \( B \) 中。如果存在，则将其加入结果集合。  时间复杂度，其中和 \( |B| \) 分别是集合和集合的大小。  2. 排序加双指针方法：  先对两个集合进行排序，然后使用双指针同时遍历两个排序后的集合。  如果当前指针所指的元素相等，则将该元素加入结果集合，并移动两个指针。  如果当前指针所指的元素不相等，移动较小元素的指针。  时间复杂度：。  3. 哈希表方法：  使用哈希表存储其中一个集合的元素，然后遍历另一个集合，对于每个元素检查是否在哈希表中。  如果存在于哈希表中，则将其加入结果集合。  时间复杂度：平均情况。  在本实验中，我们选择使用哈希表方法来实现，因为这种方法在平均情况下具有较好的时间复杂度。 | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 【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   |  | | --- | | FindIntersection.m | | function C = FindIntersection(A, B)      H = dictionary('KeyType', 'int32', 'ValueType', 'int32');      for i = 1:length(B)          H(B(i)) = 1;      end      C = [];      for i = 1:length(A)          if isKey(H, A(i))              C = [C, A(i)];          end      end  end |  |  | | --- | | Input.m | | % A = randi(100,1,10);  % B = randi(100,1,10);  A = [1 5 7];  B = [1 8 7];  C = FindIntersection(A,B) |     得到了正确的结果 |

|  |
| --- |
| 【小结与讨论】  在本次实验中，我们成功地使用MATLAB语言实现了求两个集合交集的算法。通过实验，我们实现并验证了几种不同方法来求解集合交集，特别是使用了MATLAB内置的dictionary函数和手动实现的哈希表方法。  通过本次实验，我们不仅掌握了集合交集的相关理论和实现方法，也积累了MATLAB编程的实战经验。 |