

专业课程实验报告

课程名称：算法分析与设计

开课学期：2023 至 2024 学年 第 一 学期

专业： 软件工程 年级班级：1

学生姓名： 雷艺湧 学号：222021321062002

实验教师：苟建平

计算机与信息科学学院 软件学院

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验项目名称 | | lab1 | | | |
| 实验时间 | | 年 月 日 | 实验类型 | | □验证性 □设计性 □综合性 |
| 一、实验目的  用分治策略写出并实现二分检索算法  二、实验要求  1.选择合适的数据结构来表示问题中的数列  2.根据分治策略写出分治策略的伪代码  3.使用高级语言实现伪代码  4.运行程序，验证算法正确性，并分析算法的时间复杂度 | | | | | |
| 三、实验内容与设计（主要内容，操作步骤、算法描述或程序代码）  伪代码：  5E048ED700CC98505991287833391E30 | | | | | |
| （续前表）  实际代码：  def CreateArray(x):  array = []  for i in range(x):  array.append(random.randint(0, x))  array.sort()  return array   def search(array, aim):  left, right = 0, len(array) - 1  while left <= right:  mid = left + (right - left) // 2  if array[mid] == aim:  return mid  elif array[mid] < aim:  left = mid + 1  else:  right = mid - 1   def main():  x = 1000  array = CreateArray(x)  aim = random.randint(0, x)  print(f'我们的目标是{aim}')  start = time.time()  search(array, aim)  end = time.time()  print('使用分治算法: ', end - start)  start = time.time()  for i in range(x):  if array[i] == aim:  break  end = time.time()  print('不使用分治算法: ', end - start)   if \_\_name\_\_ == '\_\_main\_\_':  main() | | | | | |
| 四、测试数据和执行结果 （在给定数据下，执行操作、算法和程序的结果，可使用数据、图表、截图等给出）  运行结果：  2023-09-21 20-51-46 的屏幕截图 | | | | | |
| 1. 实验结果分析及总结（对实验的结果是否达到预期进行分析，总结实验的收获和存在的问题等）   实验成功，使用分治策略带来了性能提升，搜索时间减少。  我是数组来表示数据结构，因为数组简单好用，而且支持随机访问，在知道索引的情况下，时间复杂度为1。  代码分析：分治策略主张把大问题转换为小问题，在二分查找中，体现为遍历一遍大数组消耗太大，那就把大数组拆分为左右两个等大的小数组进行遍历。如果数组的中间值等于目标值，那就直接返回True，表示找到了这个目标值；如果数组的中间值大于这个目标值，那就意味着目标值在中间值左侧部分的小数组，那使用递归方法调用自己去遍历左半边数组；反之，目标值就在右边。  实验收获：一开始，我简单的把数组设置成了从0-100的简单数组，结果导致分治策略一直慢于直接遍历。经过我的仔细思考后，我觉得可能是因为python的range在遍历简单数组时更有优势，于是我修改了生成数组的代码，让他随机生成在排序，结果使用分治策略就快于直接遍历。原因我想与python代码中range的底层实现有关，我查询之后发现python的range使用的是懒加载，不需要提前分配一个完整的整数序列，节省了空间；编译器对range对象也做了优化；此外，由于我原先的分治策略使用了数组切割，这样造成了额外的开销，后来我修改了代码，不用切割数组，而是改变数组的头和尾，也带来了一部分提升。  分治策略实现二分查找的时间复杂度是：logn | | | | | |
| 教  师  评  阅 | 实验内容和设计（A-E）： | | |  | |
| 操作过程、算法或代码（A-E）： | | |  | |
| 实验结果（A-E）： | | |  | |
| 实验分析和总结（A-E）： | | |  | |
| 实验成绩（A-E）：  反馈评语： | | | | |