信息学院

**实验报告**

|  |  |
| --- | --- |
| 专 业： | 计算机科学与技术 |
| 课程名称： | 数据结构 |
| 班 级： |  |
| 学 号： |  |
| 姓 名： |  |
| 辅导教师： | 赵海霞 |

2024年9月制

成绩：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实验时间 | 2024年 9 月 23 日 14 时至 16时 | 学时数 | 4学时 | |
| 组内成员 |  | | | |
| 分工情况 |  | | | |
| 1.实验名称  线性表的基本操作及应用 | | | | |
| 2.实验目的  1、熟悉C语言环境的使用以及程序的输入，编辑，调试，运行的全过程；  2、掌握线性表的顺序结构/链式结构的操作特点；  3、掌握线性表的顺序结构/链式结构的存储和常见操作在C/C++语言环境中的实现方法； | | | | |
| 3.实验内容  **A.基础型实验:**  **实现线性表的基本操作（顺序结构/链式结构下）**  （初始化，销毁、清空、判空、求长度、查找某个位置的元素、定位某个元素的位置、插入、删除、求前驱元素，求后继元素，遍历输出线性表。）  **B.提升型实验:**  **（1）验证题：对比验证线性表添加数据项的方法**  请编写程序，验证向一个线性表添加数据项的两种方法，复杂度对比：  Insert(0，,L.elem):添加数据项成为线性表的第一个元素；  append(L.elem)：添加数据项成为线性表的最后一个元素；  数据列表的规模为10万起步，步长10万；  操作1000次；  取10个数据点，打印操作时间的对比数据。  提交代码和输出结果的截屏。  格式样式如下：   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | 规模 | 10万 | 50万 | 100万 | ...... |  |  | | 时间 |  |  |  |  |  |  |   以这些数据，生成散点图  **（2）应用题：手机通讯录设计与实现**  利用基础实验的基本操作，设计手机通讯录，包含：联系人姓名、电话号码。  实现创建、插入、删除、显示、查找、求人数等功能。  **C.挑战题实验：**  **挑战题1：**有一个游戏是这样玩的，给出一列数字（环形），首先由你在其中任意选择一个数字，然后轮到你的对手选择数字，但他只能从你上一步选择的数字的左边或右边选择一个，之后你也只能从对手上一步选择的数字的左边或右边去选择，直到大家挑选完所有的数字。最后看谁选择的数字总和最大，谁就获胜。  如果一直让你第一个挑选数字，那么你能保证无论有多少个数字，数字怎么排列，你总能赢得比赛吗？  **挑战题2：狐狸逮兔子实验.**围绕着山顶有10个圆形排列的洞，狐狸要吃兔子，兔子说：“可以，但必须找到我，我就藏身于这十个洞中，你先到１号洞找，第二次隔１个洞（即3号洞）找，第三次隔２个洞（即6号洞）找，以后如此类推，次数不限。”但狐狸从早到晚进进出出了1000次，仍没有找到兔子。问兔子究竟藏在哪个洞里？  实验要求：  (1) 程序要添加适当的注释，程序的书写要采用缩进格式。  (2) 程序要具在一定的健壮性，即当输入数据非法时，程序也能适当地做出反应，如插入删除时指定的位置不对等等。  (3) 程序要做到界面友好，在程序运行时用户可以根据相应的提示信息进行操作。  (4) 根据实验报告模板详细书写电子实验报告。 | | | | |
| 4.**实验步骤**（包括结构类型定义、设计思想与实现步骤）  (1) 首先明确我们需要使用哪一种链表: 根据A.基础实验中的内容中求前驱和后继的需求可以确定使用双向链表实现会较为高效, 而根据C.挑战题中的要求, 我们可能需要循环链表的特性, 所以我们确定使用循环双向链表.  (2) 然后定义循环双向链表的结点, 循环双向链表具有prior和next两个指针分别指向前驱和后继结点, 便于寻找该结点的前驱和后继结点. 除了指针域之外还有数据域, 我们使用了泛型的思想创建了T类型模板, 使得我们的API可以适用的范围更广, 避免了重复作业.  (3) 接下来定义循环双向链表类, 为了便于后面函数的编写, 我们采用了具有头节点和尾结点的循环双向链表, 这将使得在链表头部和尾部的插入与删除的函数写的更加简洁. 类中根据A题中要求和B题中的要求声明了对应的函数, 我们将所有API封装在一个类中, 并封装为CircularLinkLis.h, 供组内其他成员统一调用, 为之后题目的解决带来了便利.  (4) 下一步我们对已完成的API进行了测试, 使用test.cpp文件, 我们对循环双链表中实现的函数进行了测试, 并对关键函数的边界情况进行了测试 如插入删除查询操作的out\_of\_range 异常. 根据测试的结果我们重新更新了CircularLinkLis.h, 使其具有更加完备的异常处理机制, 并更新了commit.  (5) 确保接口功能完善且健壮之后我们与负责实现手机通讯录的组员进行了merge, 由于我们在最初就已经声明了所需的各类函数并且提供了函数的空实现, 负责实现手机通讯录的组员只需专注于实现通讯录业务逻辑而无需重复实现循环双链表, 加快了我们完成的进度.  (5) 挑战题(1)涉及到最优的决策, 我们考虑使用动态规划找出最优解. 首先将构建的循环链表转换成数组便于更新状态, 然后根据总结出的状态转移方程求出最优解.  (6) 挑战题(2)使用了暴力求解, 从不同情况出发通过循环找到狐狸和兔子相等的洞.我们首先创建了一个包含10个洞的循环链表来表示这些洞。每个洞被编号，代表兔子可能藏匿的位置。接着，我们定义了兔子可能藏在的洞的编号。为了找到兔子，我们编写了一个名为 `findRabbit` 的函数，它将根据兔子所在的洞编号来模拟狐狸的寻找过程。  狐狸的寻找策略是按照一定的规则进行跳跃：每次跳跃的步数会逐渐增加，即第一次跳1步，第二次跳2步，以此类推。如果在跳跃过程中到达了链表的末尾，狐狸会从链表的开头重新开始跳跃，以确保能够覆盖所有洞。  在寻找过程中，我们会检查狐狸当前所在的洞是否与兔子所在的洞编号相同。如果相同，说明狐狸找到了兔子，此时我们会输出找到兔子的信息，包括尝试的次数和兔子所在的洞编号。如果在1000次尝试之后狐狸仍然没有找到兔子，我们会输出未找到兔子的信息。  通过这种方式，我们可以模拟狐狸在不同兔子藏匿位置下的寻找过程，并观察狐狸是否能够成功找到兔子。 | | | | |
| **5.调试过程及实验结果**  （详细记录在调试过程中出现的程序测试数据的选择；记录执行的结果。截图形式呈现））。  由于我们统一了接口, 所以我们只需要对CircularLinkLis.h的函数进行测试, 注意到插入删除查询通常是有可能发生超出范围的错误的, 所以我们设计了test.cpp来进行测试, 在该文件中我们设计的测试数据如下:  // 测试插入位置异常  list.insert(114514, 1919810);  // 测试获取元素异常  cout << "第233个元素为: " << list.getElement(233) << endl;  // 测试删除元素异常  list.remove(233);  int position = list.locateElement(2887);  if (position != -1)  {  cout << "元素2887的位置: " << position << endl;  }  else  {  cout << "未找到元素2887" << endl;  }  抛出异常如下:    C(1)的结果如下  C(2)的结果如下  文本  描述已自动生成 | | | |
| 6.总结（对上机实践结果进行分析，问题回答，上机的心得体会及改进意见）。  (1). 在A中, 我们构建了循环双向链表, 并使用了类模板提高了程序的泛用性, 完善了异常处理机制, 增强了程序的健壮性. 良好的API为后面的问题减少了开发时间  (2). 在B(1)中, 我们使用了chrono库对我们的程序进行了计时, 由于库函数在Windows下较差的支持, 我们通过实验多次去掉显然与现实不符的结果, 将正常结果取平均, 画出了散点图.    散点图如下    (3). 在B(2)中, 我们实现了一个基于控制台界面的通讯录, 在调用之前已经完善的API的基础上, 经过我们的测试目前没有发现异常  (4) 在C(1)中, 我们使用了动态规划算法解决问题, 能找出问题的最优解, 但是测试样例过于单薄, 还有较大的不足  问题回答如下  图形用户界面, 应用程序  描述已自动生成  (5) 在C(2)中, 我们通过程序的设计找出了不同情况下的兔子位置.  问题回答如下  文本  描述已自动生成  (6) 上机心得:  A:在构建循环双向链表的过程中，我深刻体会到了类模板的强大功能，它提高了程序的泛用性，使得我们可以轻松地处理不同类型的数据。同时，通过完善异常处理机制，我学会了如何增强程序的健壮性，确保在遇到错误输入或操作时程序能够优雅地处理，而不是崩溃。此外，良好的API设计大大减少了后续开发的时间，提高了开发效率。  B(1)在验证线性表添加数据项的方法时，我学习到了如何通过实验来验证算法的性能。通过编写程序并记录操作时间，我了解到了如何通过实验来验证算法的性能，并且学会了如何使用散点图来直观地展示数据。  B(2)在实现基于控制台界面的通讯录时，我将之前完善的API应用到了实际问题中。这个过程让我理解了如何将理论知识转化为解决实际问题的工具。经过我们的测试，目前没有发现异常，这增强了我对程序稳定性的信心。  C(1)在解决这个问题时，我使用了动态规划算法来找出问题的最优解。这个过程让我深刻理解了动态规划的强大能力，但同时也意识到了测试样例的不足，这提示我在未来的工作中需要设计更全面的测试样例来确保算法的健壮性。  C(2)在设计程序找出不同情况下的兔子位置时，我不仅提高了我的编程技能，还增强了我的逻辑思维和问题解决能力。通过解决这个问题，我学会了如何将数学问题转化为编程问题，并找到了有效的解决方案。  (7) 改进意见:  提供问题的测试样例与答案的示例 | | | |
| **7.附录**（源程序代码等）  代码略长, 请见压缩包中的src文件夹 | | | |