一、实验目的

掌握数据结构相关概念、算法时间复杂度和空间复杂度的计算运用。

二、实验内容

1. 简述下列概念：数据、数据元素、数据项、数据对象、数据结构、逻辑结构、

存储结构、抽象数据类型。

- **数据**：指能被计算机识别、存储和处理的**各种符号的总称**，是计算机程序加工的 “原材料”。
- **数据元素**：数据的**基本单位**，也称 “记录”，是计算机程序中直接处理的对象。
- **数据项**：数据的**最小不可分割单位**，用于描述数据元素的某一具体属性。
- **数据对象**：由**性质相同的数据元素组成的集合**，是数据的一个子集。
- **数据结构**：相互之间存在一种或多种特定关系的数据元素的集合
- **逻辑结构**：指数据元素之间**抽象的逻辑关系**，仅描述元素之间的关联方式，与计算机的存储硬件无关。
- **存储结构**：也称 “物理结构”，指逻辑结构在计算机**内存中的具体存储方式**（即如何将元素及元素间的关系映射到物理存储单元）。
- **抽象数据**：指一个**数学模型**以及定义在该模型上的一组运算，它仅关注 “做什么”（逻辑功能），不关注 “怎么做”（具体实现）。

2. 举一个数据结构例子，叙述其逻辑结构和存储结构两方面的含义和相互关

系。
以单链表为例：
单链表的逻辑结构：
逻辑结构是**数据元素之间抽象的逻辑关系**，与数据在计算机中的物理存储位置无关，仅描述 “元素如何关联”。单链表的逻辑结构属于**线性结构**，其核心特征是：

- 数据元素（称为 “节点”）之间存在 **“一对一” 的线性关系** ；
- 除第一个节点（头节点）外，每个节点有且仅有一个**前驱节点**；
- 除最后一个节点（尾节点）外，每个节点有且仅有一个**后继节点**。
- 其实我觉得就和糖葫芦是一个道理，每个山楂就是一个节点，山楂之间用竹签链接，从first到last就是一个线性的序列
  单链表的存储结构：

存储结构（物理结构）是**逻辑结构在计算机内存中的具体实现方式**，即 “如何将抽象的逻辑关系映射到物理存储单元”，需解决 **“元素存哪里”** “**关系怎么体现**” 两个问题。

- 数据元素（节点）存储在**非连续的内存地址**中（内存中可分散存放）；
- 每个节点包含两部分：
  ① **数据域**：存储节点本身的具体数据（如学生的姓名、学号）；
  ② **指针域（链域）**：存储**后继节点的内存地址**（通过指针体现 “前驱→后继” 的逻辑关系）。

逻辑结构与存储结构是 **“抽象设计” 与 “物理实现” 的依存关系**

- 逻辑结构是存储结构的 “设计依据”
  逻辑结构定义了数据元素的关系本质，决定了存储结构的设计方向 —— 存储结构必须 “准确体现逻辑关系”，否则数据无法被正确处理。
- 存储结构是逻辑结构的 “物理载体”
  抽象的逻辑关系无法直接被计算机处理，必须通过存储结构落地到物理内存中 —— 存储结构将 “逻辑关系” 转化为 “可被硬件识别的物理关联”。
- 逻辑结构不变，存储结构可灵活选择（但需匹配逻辑）
  同一种逻辑结构可以对应多种存储结构，选择哪种存储结构取决于 “运算效率需求”（如插入、删除、查找的频率），但无论哪种存储结构，都必须忠实体现逻辑关系。

3. 分析下列算法的时间复杂度。

(1)设n 是描述问题规模的非负整数，计算下面程序段的时间复杂度：

```
x=2;
while(x<n/2)
x=2*x;
```

结果是 **O(log n)**
思路：

1. 初始时 $x = 2$。
2. 每次循环，x 变为 $2x$。
3. 假设循环执行 k 次，则有：

$$2 \cdot 2^k \approx \frac{n}{2}$$

4. 化简可得：

$$2^{k+1} \approx \frac{n}{2}$$

$$2^k \approx \frac{n}{4}$$

5. 两边取对数：

$$k \approx \log_2 \frac{n}{4}$$

6. 拆开常数项：

$$\log_2 \frac{n}{4} = \log_2 n - 2$$

   当 n 足够大时，常数 -2 可以忽略。

$$T(n) = O(\log n)$$

(2) 设n 是描述问题规模的非负整数，计算下面程序段的时间复杂度：

```
x=0;
while(n>=(x+1)*(x+1))
x=x+1;
```

1. 初始时 $x = 0$。
2. 每次循环，x 增加 1。
3. 循环条件是：

$$n \geq (x+1)^2$$

   即：

$$(x+1)^2 \leq n$$

4. 当循环结束时，$x \approx \sqrt{n}$。
   也就是说，循环次数大约为 $\sqrt{n}$。

$$T(n) = O(\sqrt{n})$$

三、实验环境

Windows 操作系统，Visual C 程序集成环境或 Visual studio 程序集成环境。

四、实验总结

这个是基本的概念 作为了解以为了之后深入数据结构的深入学习是有帮助于进行算法学习和打算法比赛的。重要的原因我觉得有以下几点：

- **避免 TLE（超时）**：能在看题时快速判断某个解法是否可行，避免花时间写注定会超时的方案。

- **快速选算法**：读到输入规模就能立刻缩小候选算法（暴力、贪心、二分、动态、图算法等）。
- **权衡实现难度与可行性**：在比赛里常常要在「能跑得快」和「能在短时间实现」之间取舍。
- **定位性能瓶颈**：知道复杂度后更容易查出哪段代码要优化（数据结构？循环嵌套？重复计算？）。
- **正确估分/调策略**：决定先做哪题（容易取分的 O(n) 题 vs 难的 O(n log n) 题）。