

数据结构实验报告

你的名字

2025 年 6 月 1 日

1 引言

1.1 实验背景与目的

数据结构是计算机科学中的核心内容，它直接影响算法的效率和程序的性能。本次实验的目标是实现和测试线性表、树、图的基本操作，以加深对这些数据结构的理解。

1.2 实验环境与工具

- 编程语言：C++/Python
- 开发环境：Visual Studio Code/PyCharm
- 测试工具：单元测试框架（如 Google Test）

2 线性表的实现与测试

2.1 线性表的基本概念

线性表是一种有序的元素集合，支持插入、删除、查找等操作。

2.2 线性表的实现

使用数组实现顺序表，使用链表实现单链表。以下是关键操作的伪代码：

// 插入操作伪代码

```
void insert(List &list, int index, ElementType value) {
```

操作类型	输入数据	预期输出	实际输出
插入	位置 1, 值 10	成功	成功
删除	位置 2	成功	成功
查找	值 10	位置 1	位置 1

表 1: 线性表测试结果

```

if (index < 0 || index > list.length) {
    throw "Index out of bounds";
}
for (int i = list.length; i > index; i--) {
    list.data[i] = list.data[i - 1];
}
list.data[index] = value;
list.length++;
}

```

2.3 线性表的测试

设计测试用例并分析测试结果，如下表所示：

3 树的实现与测试

3.1 树的基本概念

树是一种层次结构，支持插入、删除、遍历等操作。

3.2 树的实现

使用链表实现二叉树。以下是关键操作的伪代码：

```

// 插入操作伪代码
void insert(TreeNode &root, int value) {
    if (root == null) {
        root = new TreeNode(value);
    } else if (value < root.value) {

```

操作类型	输入数据	预期输出	实际输出
插入	值 5	成功	成功
删除	值 5	成功	成功
遍历	中序	1,2,3	1,2,3

表 2: 树测试结果

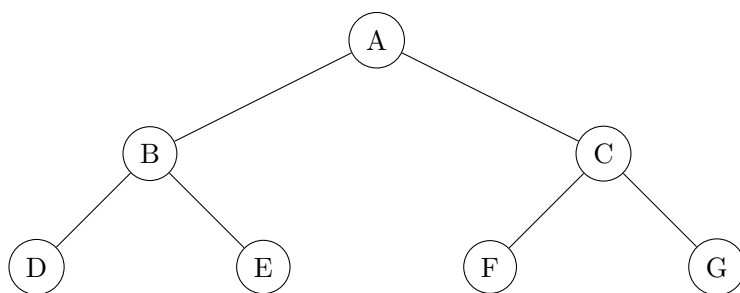


图 1: 二叉树的结构图

```

        insert(root.left, value);
    } else {
        insert(root.right, value);
    }
}

```

3.3 树的测试

设计测试用例并分析测试结果，如下表所示：

4 图的实现与测试

4.1 图的基本概念

图是一种由顶点和边组成的结构，支持插入顶点、插入边、遍历等操作。

4.2 图的实现

使用邻接矩阵和邻接表实现图。以下是关键操作的伪代码：

操作类型	输入数据	预期输出	实际输出
插入顶点	顶点 A	成功	成功
插入边	边 A-B	成功	成功
遍历	深度优先	A,B,C	A,B,C

表 3: 图测试结果

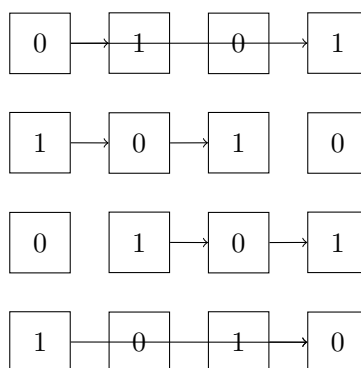


图 2: 邻接矩阵示意图

```
// 插入顶点操作伪代码
void insertVertex(Graph &graph, VertexType vertex) {
    if (graph.vertexCount >= MAX_VERTEX) {
        throw "Graph is full";
    }
    graph.vertices[graph.vertexCount++] = vertex;
}
```

4.3 图的测试

设计测试用例并分析测试结果，如下表所示：

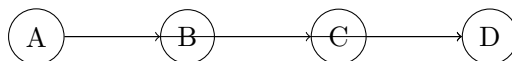


图 3: 邻接表示意图

5 实验总结

5.1 实验结果总结

通过本次实验，我们成功实现了线性表、树、图的基本操作，并进行了详细的测试。

5.2 实验中的问题与解决方案

在实现过程中，我们遇到了一些问题，如内存管理、边界条件处理等，通过调试和优化，最终解决了这些问题。

5.3 未来工作展望

未来可以进一步优化数据结构性能，并扩展更多功能，如支持并发操作、动态调整大小等。

6 参考文献

- 《数据结构与算法分析》 - Mark Allen Weiss
- 《算法导论》 - Thomas H. Cormen

7 附录

7.1 源代码

提供完整的源代码（线性表、树、图的实现）。

7.2 测试数据

提供测试用例的详细数据。