**作业4 图型结构及其应用**

1. **分别实现无向图的邻接矩阵和邻接表存储结构的建立算法，分析和比较各建立算法的时间复杂度以及存储结构的空间占用情况。**

**为简化后续程序，本作业将邻接矩阵和邻接表储存在同一结构体中，具体如下：**文本

描述已自动生成

**其中，struct vertexNode为邻接表的表头数组，定义如下：**

文本

中度可信度描述已自动生成

EdgeNode为后续连接的节点，定义为：

文本

中度可信度描述已自动生成

建立思路为输入顶点个数以及边数后，逐边对**邻接矩阵和邻接表进行修改或者插入，**具体建立过程如下：

1. **无向图的邻接矩阵的建立**

文本

描述已自动生成

文本

描述已自动生成

1. **无向图的邻接表的建立**

文本

描述已自动生成

由程序可得，设为图所含边数，为顶点数目，无论邻接表或是邻接矩阵的建立时间复杂度均为,但是邻接表的空间复杂度为，而邻接矩阵的空间复杂度为。

1. **实现无向图的邻接矩阵和邻接表两种存储结构的相互转换算法。**

转换过程和建立过程相似，由邻接表或邻接矩阵读出边信息后再存入对应的矩阵或者表中即可，具体程序如下：

1. 邻接表转换为邻接矩阵

文本

描述已自动生成

1. 邻接矩阵转换为邻接表

文本

描述已自动生成

由程序可知，设为图所含边数，为顶点数目，则邻接表转换为邻接矩阵的时间复杂度为，邻接矩阵转换为邻接表的时间复杂度为。

1. **在上述两种存储结构上，分别实现无向图的深度优先搜索(递归和非递归)和广度优先搜索算法。并以适当的方式存储和展示相应的搜索结果，包括：深度优先或广度优先生成森林（或生成树）、深度优先或广度优先序列和深度优先或广度优先编号。并分析搜索算法的时间复杂度和空间复杂度。**
2. **邻接表**
3. **深度优先搜索(递归)**

**屏幕上有字

描述已自动生成**

**文本

描述已自动生成**

1. **深度优先搜索(非递归)，基于栈结构**

**文本

描述已自动生成**

**文本

描述已自动生成**

1. **广度优先搜索，基于队列结构**

**文本

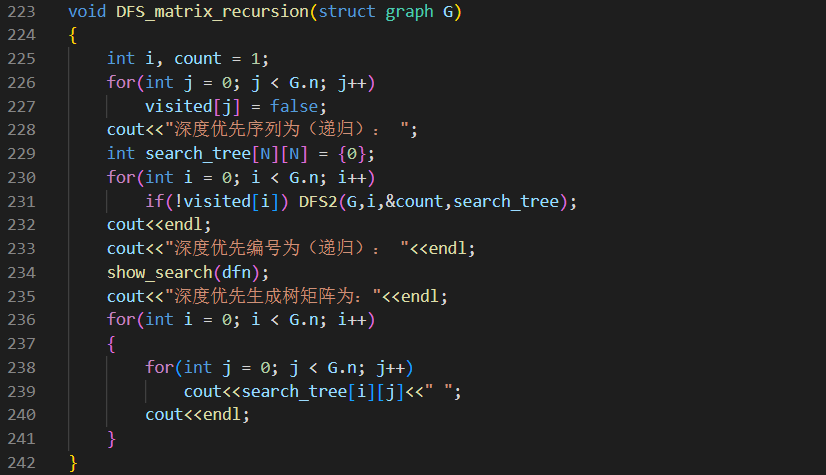
描述已自动生成**

**文本

描述已自动生成**

由程序可得，设为顶点数，为边数，则对邻接表进行深度（广度）优先搜索的时间复杂度为，对每个顶点都访问一次的时间复杂度为，而图的遍历过程是对每个顶点通过边查找邻接顶点的过程，这一过程开销为,而空间复杂度为，这是因为最坏的情况下所有的顶点进入到栈或队列当中。

1. **邻接矩阵**
2. **深度优先搜索(递归)**

****

**文本

描述已自动生成**

1. **度优先搜索(非递归)，基于栈结构**

**文本

描述已自动生成**

**电脑的屏幕截图

描述已自动生成**

1. **广度优先搜索，基于队列结构**

文本

描述已自动生成

由程序可得，则对邻接表进行深度（广度）优先搜索的时间复杂度为,因为需要对矩阵中每个位置进行遍历，而空间复杂度为，这是因为最坏的情况下所有的顶点进入到栈或队列当中。

1. **对于无向图，采用“邻接表”存储结构，设计和实现计算每个顶点度的算法，并分析其时间复杂度。**

只需对每个顶点之后连接的顶点数进行遍历即可，具体程序如下：文本

描述已自动生成

由程序可得，设为顶点数，为边数，则对邻接表计算度数的时间复杂度为，空间复杂度为

1. **以适当的方式输入图的顶点和边，并显示相应的结果。要求顶点不少于10个，边不少于13个。**

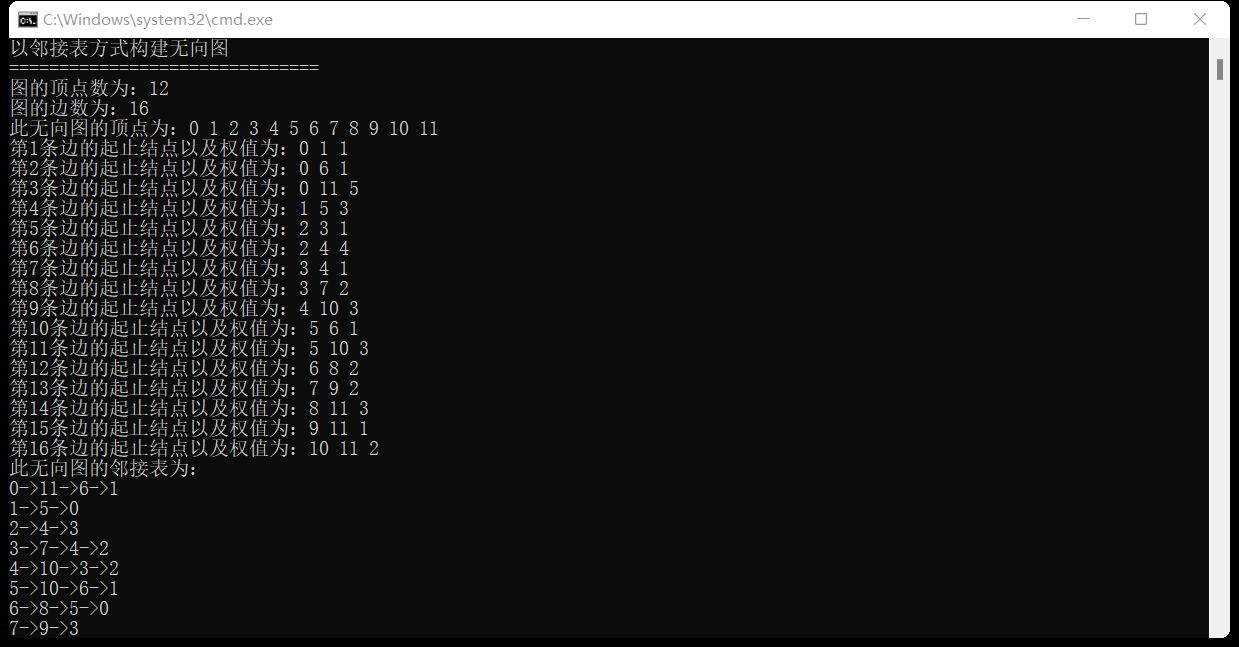
如图所示，图中共有12个顶点，16条边，包含权值等信息。为方便观察，以下分为两个程序，一个展示以邻接表形式储存图以及遍历，另外一种以邻接矩阵形式储存图以及遍历。

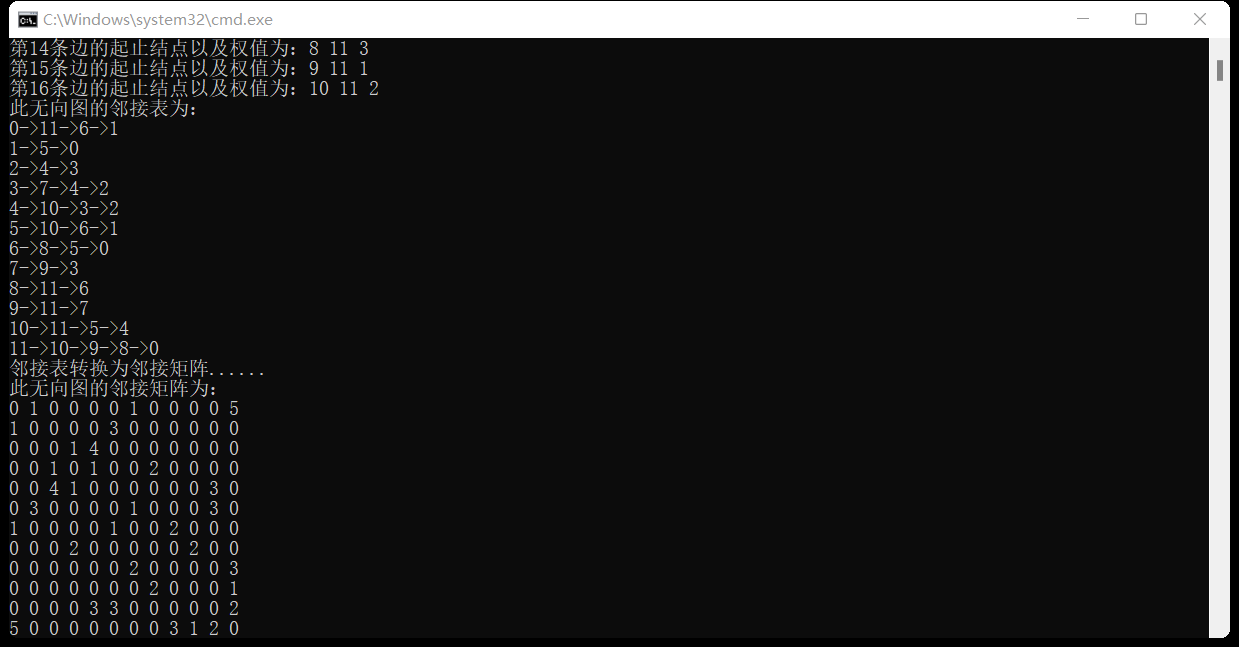
图示

描述已自动生成

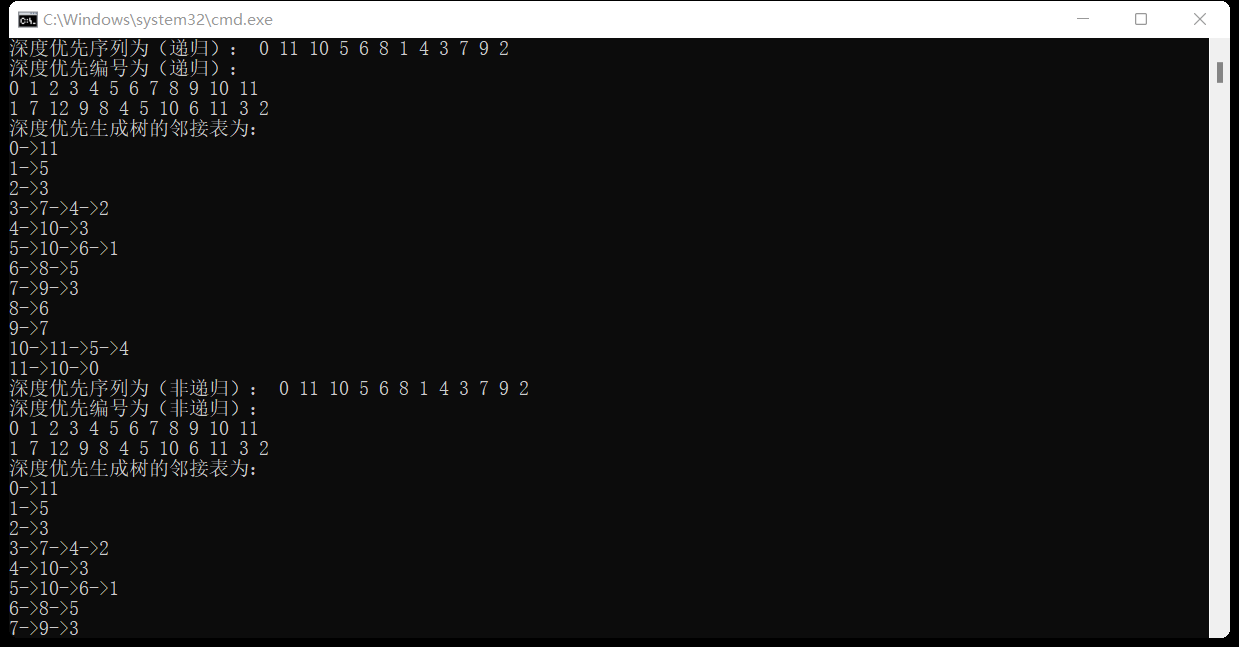
1. 以邻接表方式构建无向图：

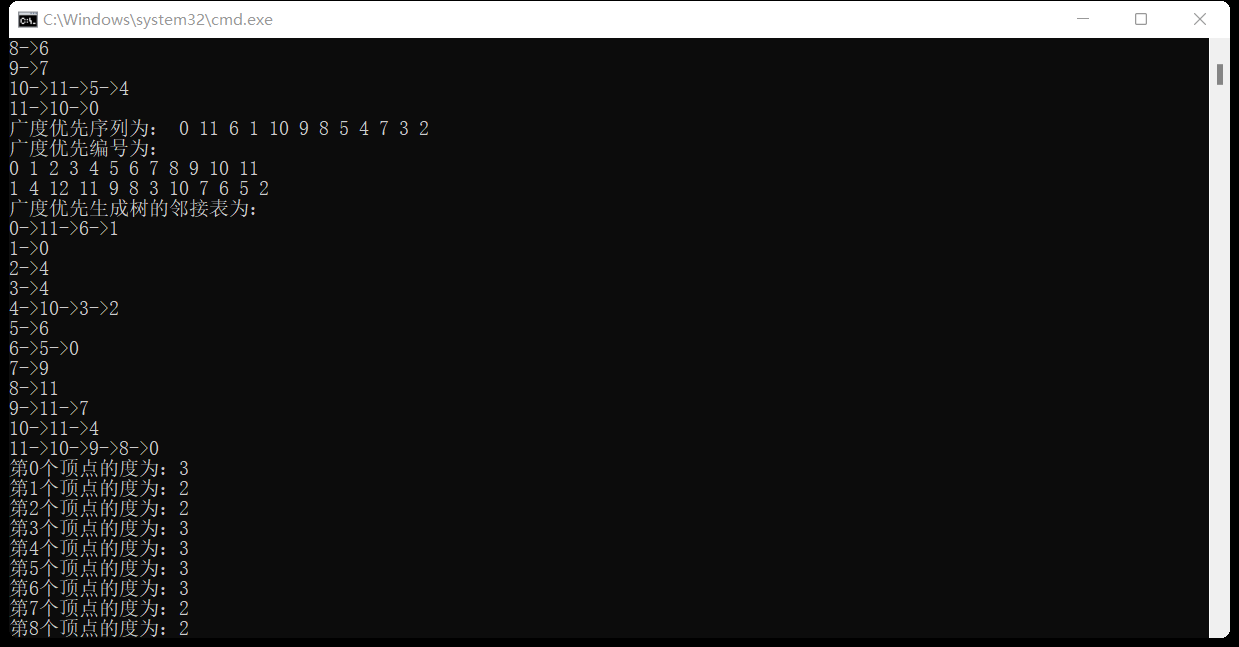
1.输入

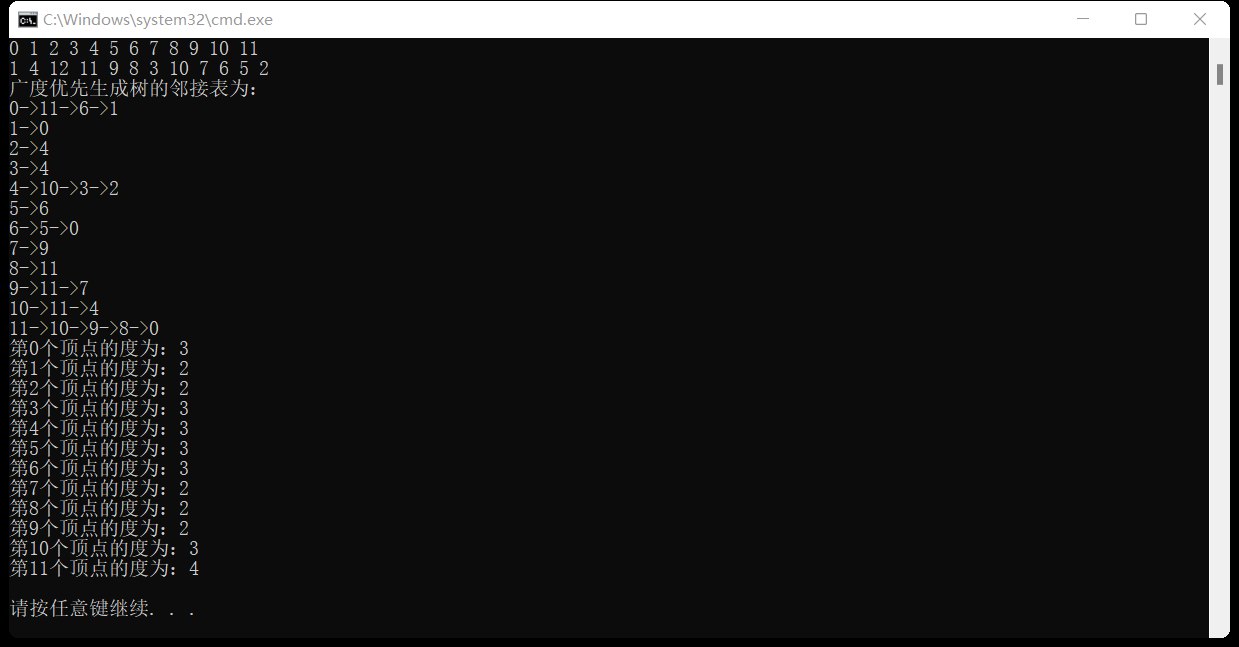


2.显示以及转换

3.搜索以及顶点度数







4.深度优先树以及广度优先树的图示图示

描述已自动生成图示

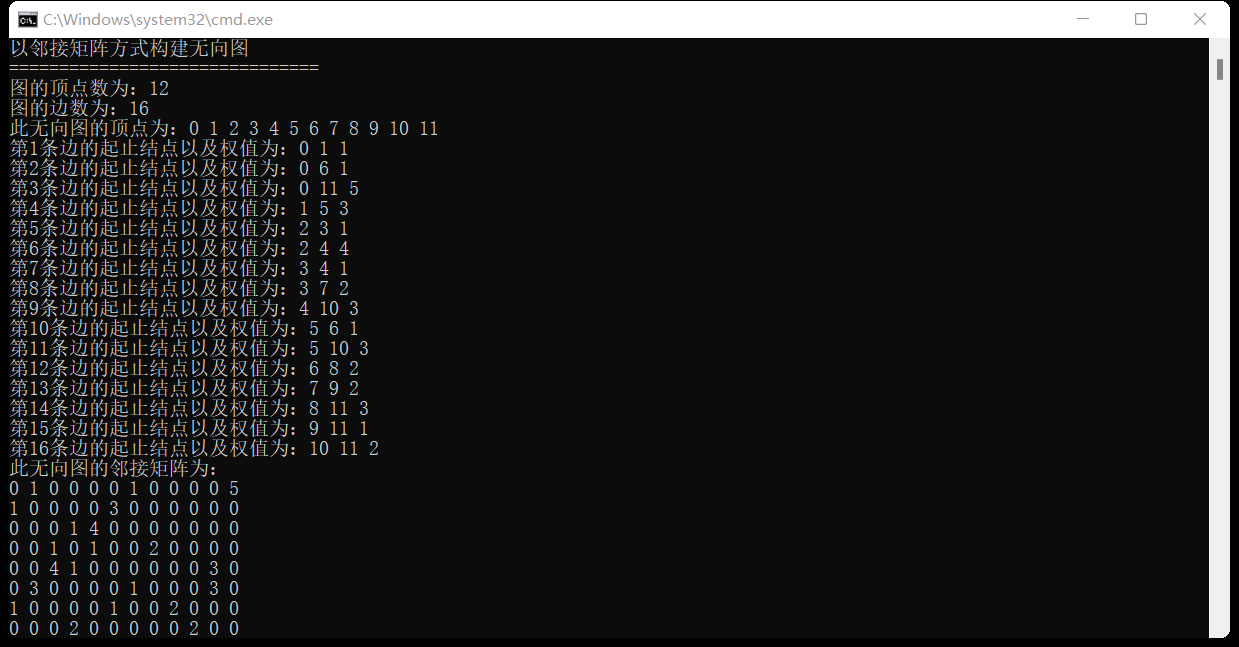
描述已自动生成

Figure 1 深度优先树

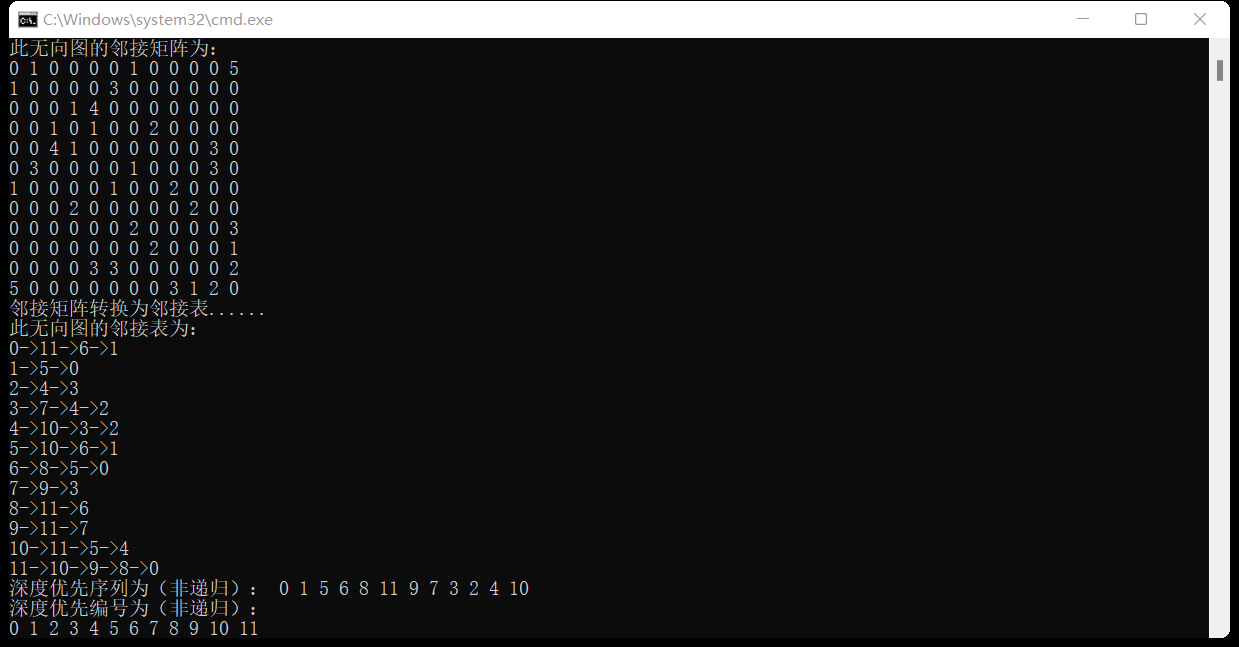
Figure 2 广度优先树

1. 以邻接矩阵方式构建无向图：

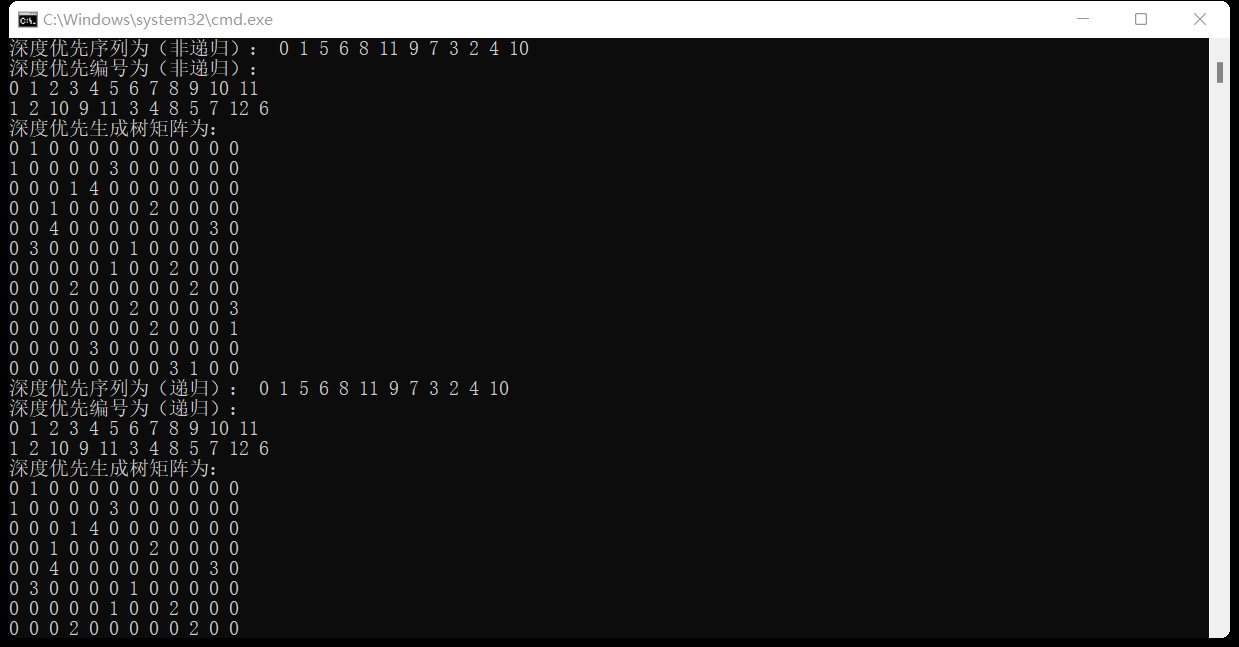
1.输入

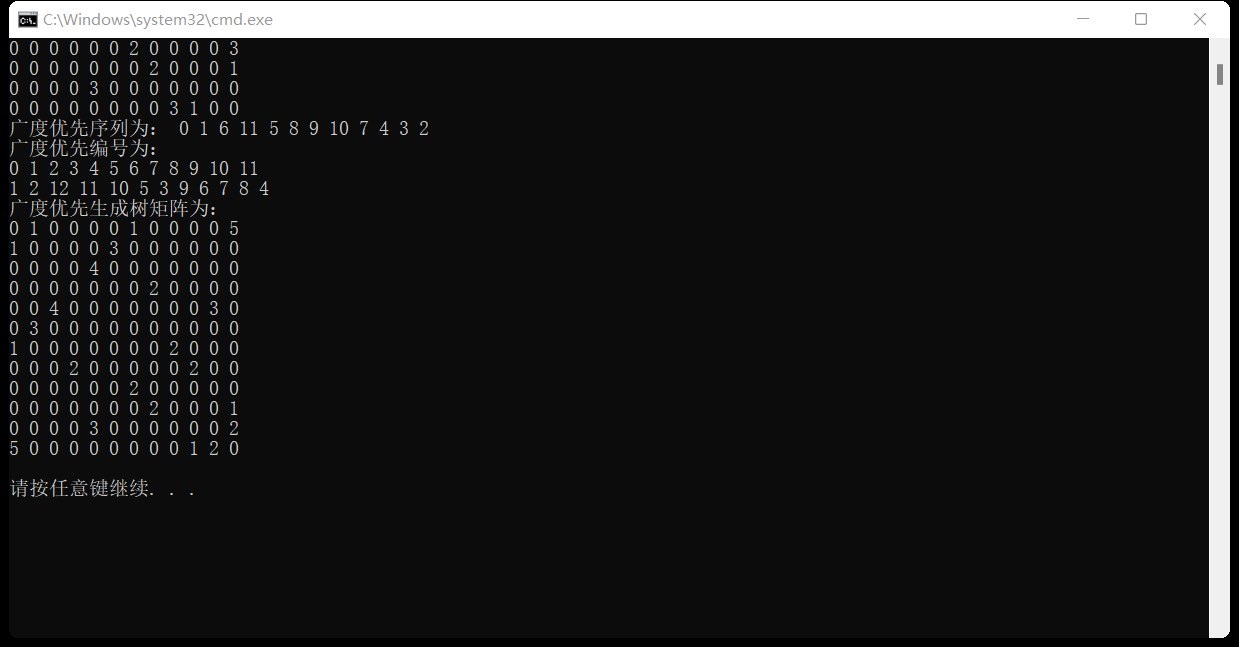


2.显示以及转换



3.搜索





图示, 示意图

描述已自动生成图片包含 图示

描述已自动生成4.深度优先树以及广度优先树的图示

Figure 4 广度优先树

Figure 3 深度优先树