实验报告

## 实验名称 ：图遍历

## 需求分析

1. 题目的意思为，按照输入建立图的临接多重表结构，然后根据存储结构，输出dfs访问序列以及相应的生成树边集和输出bfs访问序列以及相应生成树的边集。
2. 根据需要可得，分为三步：
   1. 以临接多重表建图
   2. dfs输出访问序列，再输出dfs生成树的边集。
   3. bfs输出访问序列，再输出bfs生成树的边集。

## 概要设计

在main函数中分几部分来完成它.

1.用creatGraph()函数来建立图的临接多重表存储结构;

2.用dfstra()输出dfs访问序列，并存储下生成树的边集。

3.输出dfs生成树的边集。

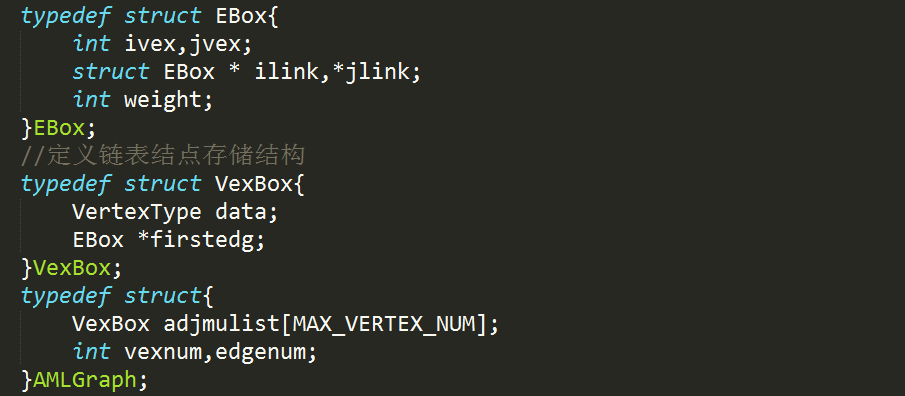
4.用bfstra()输出bfs访问序列，并存储下生成树的边集。

5.输出bfs生成树的边集。

## 详细设计.

1. creatGraph()函数的实现：

既然要建立图的临接多重表，那么得先看看它的存储结构。如下:



可见图的临接多重表变化的仅仅是存边信息的EBox.所以就按照临接表的思路来建图。

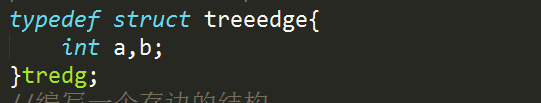
首先得得到结点的数目，以及边的数量，然后依次得到每个节点的信息，之后就是建立每一个图了。用尾插法来建立图的临接多重表存储结构。如图：

·

如上我们写好了图的临接多重表建图算法。

1. 用dfstra()输出dfs访问序列，并存储下生成树的边集

既然要存储生成树的访问边集，就得定义一个存储结构。由于临接多重表里存储的是元素在顶点存储结构的下标。故设计如下存储结构：

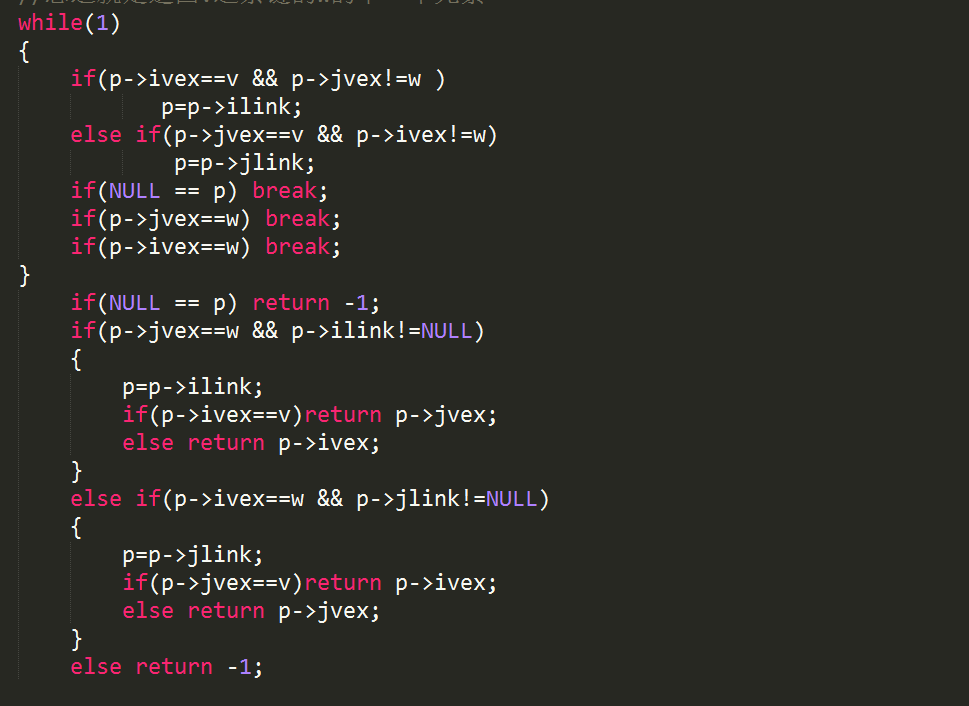


由于要需要需要用户输入起点。故，在dfstra中用户只需要得到用户输入的结点的下标x，从x之后dfs在，从x之前> 0 dfs即可得到dfs的生成序列。

那么什么时候存储生成树的边集呢？？由于在dfs的时候，需要在得到下一条边，以及需要判断当前的结点是否被访问过。那么在每次visit顶点的时候，我们只需要记录此时的其实结点以及访问的结点，待到dfs完毕之后直接输出即可。



按照以上思路,很容易编写出整体框架。但是，在获取下一条边的时候会有一个问题。我当前的firstedg一直找到w下标所在的边的时候。下一条边需要进行判断，指向的边ivex还是jvex是等于起始节点的。否则会让程序陷入死循环。具体实现如下：



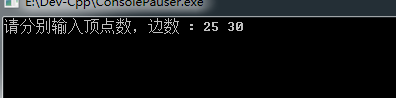
3.用bfstra()输出bfs访问序列，并存储下生成树的边集。

其实bfs和dfs的操作基本相同。唯一的区别是在遍历的时候，一个用的递归，另一个用的队列。做法一模一样，这里不做赘述了。

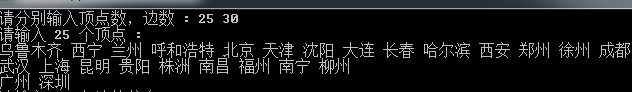
## 用户手册

**输入数据：见附录**

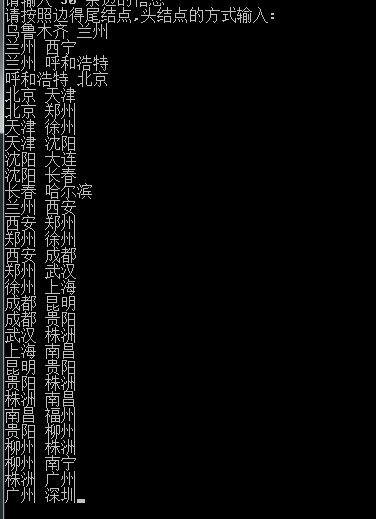
**首先在运行程序后会要求用户输入数据如下：**



之后输入如下:



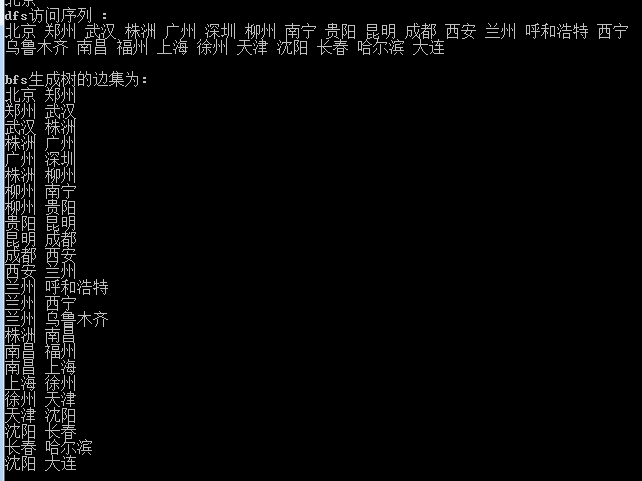
之后按要求输入：



最后输入起点：



得到结果如下:





## 附录

输入数据文件名称: in.txt

主要程序: 遍历(临接多重表).cpp