作业4.2

图的存储结构的建立与搜索

图的搜索（遍历）算法是图型结构相关算法的基础，本作业要求编写程序演示**有向图**典型存储结构的建立和搜索（遍历）过程。

存储结构

```
//邻接矩阵的存储结构
class AdjacencyMatrix
{
private:
    int Matrix[MAXN][MAXN];//邻接矩阵
    bool Visited[MAXN], Appeared[MAXN];//该节点是否访问过；该节点是否出现过
    int n, m;//节点的数量和边的数量
};
//邻接表的存储结构
class AdjacencyList
{
private:
    struct Node
    {
        int To;
        Node *Next;
        Node()
        {
            Next = NULL;
        }
    };
    Node *List[MAXN];//点集
    bool Visited[MAXN], Appeared[MAXN];//该节点是否访问过；该节点是否出现过
    int n, m;//节点的数量和边的数量
};
```

函数说明

1. 分别实现有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；

`void InputandBuild()` 建立邻接矩阵/邻接表。（该函数在邻接矩阵和邻接表的结构体中命名相同）

邻接表的建立时间复杂度：$O(N+M)$；空间占用：$O(N+M)$

邻接矩阵的建立时间复杂度：$O(N^2)$；空间占用：$O(N^2)$

2. 实现有向图的邻接矩阵和邻接表两种存储结构的相互转换算法；

`AdjacencyMatrix Transformer_List2Matrix(AdjacencyList &AL)` 将邻接表转换为邻接矩阵。

`AdjacencyList Transformer_Matrix2List(AdjacencyMatrix &AM)` 将邻接矩阵转换为邻接表。

3. 在上述两种存储结构上，分别实现有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显示相应的搜索结果（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编号）；

`Result DFSWithRecursion()` 用递归实现深度优先搜索。（该函数在邻接矩阵和邻接表的结构体中命名相同）

`Result DFSWithoutRecursion()` 用非递归实现深度优先搜索。（该函数在邻接矩阵和邻接表的结构体中命名相同）

`Result BFSWithoutRecursion()` 用非递归实现广度优先搜索。（该函数在邻接矩阵和邻接表的结构体中命名相同）

`void Print()` 在 `struct Result()` 里，输出遍历结果，包括深度优先或广度优先序列和编号（**用序列代表生成森林（或生成树）**）。

注：由于 DFS 时进入下个节点的顺序可能不相同，所以递归和非递归出来的 DFS 序也可能不相同。

4. 分析搜索算法的时间复杂度和空间复杂度；

邻接表深度优先搜索时间复杂度：$O(M+N)$；空间复杂度：$O(N)$

邻接矩阵深度优先搜索时间复杂度：$O(N^2)$；空间复杂度：$O(N)$

邻接表广度优先搜索时间复杂度：$O(M+N)$；空间复杂度：$O(N)$

邻接矩阵广度优先搜索时间复杂度：$O(N^2)$；空间复杂度：$O(N)$

5. 以文件形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 15 条；

`freopen("Homework4_In.txt","r",stdin);` 文件形式输入。

6. 软件功能结构安排合理，界面友好，便于使用。

```
printf("--------------------------------\n"
                "0.Quit\n"
                "1.Input Graph using Adjacency List\n"
                "2.Input Graph using Adjacency Matrix\n"
                "3.Transform List to Matrix\n"
                "4.Transform Matrix to List\n"
                "5.DFS List With Recursion\n"
                "6.DFS Matrix With Recursion\n"
                "7.DFS List Without Recursion\n"
                "8.DFS Matrix Without Recursion\n"
                "9.BFS List\n"
                "10.BFS Matrix\n"
                "--------------------------------\n"
                "Input an Integer to Choose:\n");
```

自测

测试说明

由于本程序未加入`system("pause")`，建议在CMD/Terminal中测试。

本程序采用输入文件输入，输出采用标准输出。

为方便测试，源程序 `Homework4.cpp` 中**将测试内容写入主函数**。

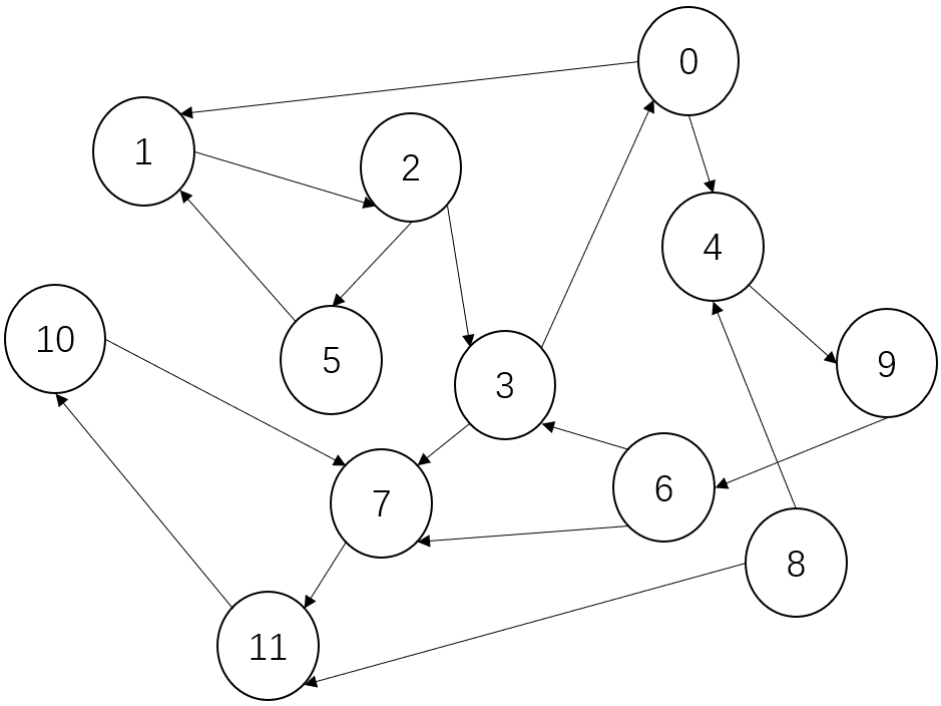
另提供样例测试数据输入 `Homework4_In.txt` ，和期望输出 `Homework4_Out.txt` 。

操作说明

> 0.Quit
> 1.Input Graph using Adjacency List
> 2.Input Graph using Adjacency Matrix
> 3.Transform List to Matrix
> 4.Transform Matrix to List
> 5.DFS List With Recursion
> 6.DFS Matrix With Recursion
> 7.DFS List Without Recursion
> 8.DFS Matrix Without Recursion
> 9.BFS List
> 10.BFS Matrix

数据说明

Homework4_In.txt




```
1                   输入图到邻接表中
12 17               12个点17个边的图
0 4                 图的边的描述
4 9
```

```
9 6
6 3
3 0
2 5
5 1
1 2
10 7
7 11
11 10
2 3
8 4
8 11
6 7
3 7
0 1
3                   邻接表转邻接矩阵
4                   邻接矩阵转邻接表
5                   在邻接表上递归地DFS
6                   在邻接矩阵上递归地DFS
7                   在邻接表上非递归地DFS
8                   在邻接矩阵上非递归地DFS
9                   在邻接矩阵上BFS
10                  在邻接表上BFS
0                   退出
```

Homework4_Out.txt

```
--------------------------------
0.Quit
1.Input Graph using Adjacency List
2.Input Graph using Adjacency Matrix
3.Transform List to Matrix
4.Transform Matrix to List
5.DFS List With Recursion
6.DFS Matrix With Recursion
7.DFS List Without Recursion
8.DFS Matrix Without Recursion
9.BFS List
10.BFS Matrix
--------------------------------
Input an Integer to Choose:             输入图到邻接表中，邻接表的描述
0 : 1 4
1 : 2
2 : 3 5
3 : 7 0
4 : 9
5 : 1
6 : 7 3
7 : 11
8 : 11 4
9 : 6
10 : 7
11 : 10
**提示（略去）**
Input an Integer to Choose:             邻接表转邻接矩阵，邻接矩阵的描述
0 1 0 0 1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0
```

```
0 0 0 1 0 1 0 0 0 0 0
1 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 1 0
0 1 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0
**提示（略去）**
Input an Integer to Choose:              邻接矩阵转回邻接表，邻接表的描述
0 : 4 1
1 : 2
2 : 5 3
3 : 7 0
4 : 9
5 : 1
6 : 7 3
7 : 11
8 : 11 4
9 : 6
10 : 7
11 : 10
**提示（略去）**
Input an Integer to Choose:              在邻接表上递归地DFS
DFSWithRecursion
Sequence:
0 4 9 6 7 11 10 3 1 2 5 8                DFS序
Node  ID                                 编号
  0 : 1
  1 : 9
  2 : 10
  3 : 8
  4 : 2
  5 : 11
  6 : 4
  7 : 5
  8 : 12
  9 : 3
  10 : 7
  11 : 6

**提示（略去）**
Input an Integer to Choose:              在邻接矩阵上递归地DFS
DFSWithRecursion
Sequence:
0 1 2 3 7 11 10 5 4 9 6 8                DFS序
Node  ID                                 编号
  0 : 1
  1 : 2
  2 : 3
  3 : 4
  4 : 9
  5 : 8
  6 : 11
  7 : 5
  8 : 12
  9 : 10
```

```
  10 : 7
  11 : 6

**提示（略去）**
Input an Integer to Choose:              在邻接表上非递归地DFS
DFSWithoutRecursion
Sequence:
0 1 2 3 7 11 10 5 4 9 6 8                DFS序
Node  ID                                 编号
  0 : 1
  1 : 2
  2 : 3
  3 : 4
  4 : 9
  5 : 8
  6 : 11
  7 : 5
  8 : 12
  9 : 10
  10 : 7
  11 : 6

**提示（略去）**
Input an Integer to Choose:              在邻接矩阵上非递归地DFS
DFSWithoutRecursion
Sequence:
0 4 9 6 7 11 10 3 1 2 5 8                DFS序
Node  ID                                 编号
  0 : 1
  1 : 9
  2 : 10
  3 : 8
  4 : 2
  5 : 11
  6 : 4
  7 : 5
  8 : 12
  9 : 3
  10 : 7
  11 : 6

**提示（略去）**
Input an Integer to Choose:              在邻接矩阵上BFS
BFSWithoutRecursion
Sequence:
0 4 1 9 2 6 5 3 7 11 10 8                BFS序
Node  ID                                 编号
  0 : 1
  1 : 3
  2 : 5
  3 : 8
  4 : 2
  5 : 7
  6 : 6
  7 : 9
  8 : 12
  9 : 4
  10 : 11
```

```
  11 : 10

**提示（略去）**
Input an Integer to Choose:              在邻接表上BFS
BFSWithoutRecursion
Sequence:
0 1 4 2 9 3 5 6 7 11 10 8                BFS序
Node  ID                                 编号
  0 : 1
  1 : 2
  2 : 4
  3 : 6
  4 : 3
  5 : 7
  6 : 8
  7 : 9
  8 : 12
  9 : 5
  10 : 11
  11 : 10

**提示（略去）**
Input an Integer to Choose:              退出
```