

## 作业 4 图形结构及其应用

班级：22WL022

姓名：杨明达

学号：2022110829

一、分别实现有向图的邻接矩阵和邻接表存储结构的立算法, 分析和比较其建立算法的时间复杂度以及存储结构的空间占用情况。

### (1) 构造有向图

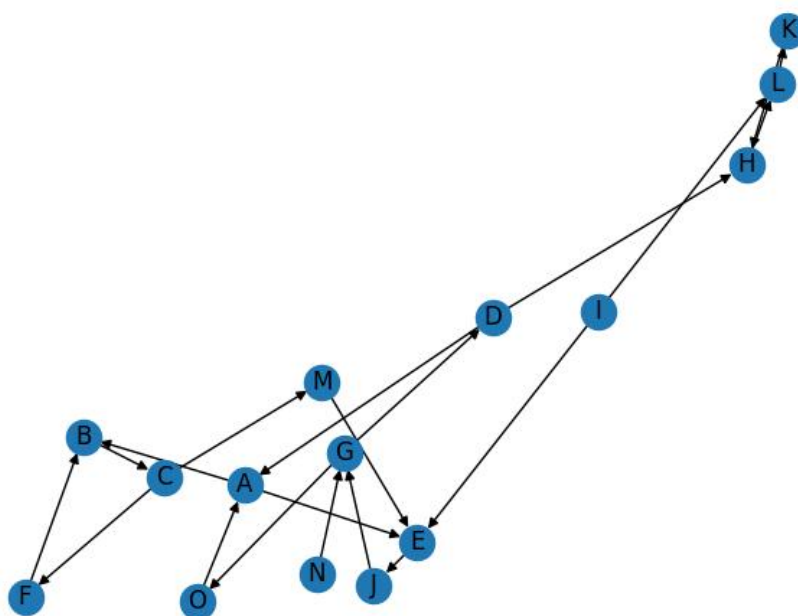


图 1 有向图

### (2) 有向图的邻接矩阵

请输入图的顶点数和边数:	图 的邻接矩阵为:
15 20	0 1 0 0 1 0 0 0 0 0 0 0 0 0 0
请输入图的顶点信息:	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
ABCDEFGHIJKLMNO	0 0 0 0 0 1 0 0 0 0 0 0 0 1 0
请输入图的边和权值:	1 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 1 1	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 4 1	0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1 2 1	0 0 0 1 0 0 0 0 0 0 0 0 0 0 1
2 12 1	0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
2 5 1	0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
3 0 1	0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
3 7 1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
4 9 1	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
5 1 1	0 0 0 0 1 0 0 0 0 0 0 0 1 0 0
6 3 1	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
6 14 1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
7 11 1	0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
8 4 1	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
8 11 1	0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
9 6 1	0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
10 7 1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
11 10 1	0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
12 4 1	0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
13 6 1	0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
14 1 1	

图 2 邻接矩阵

### (3) 有向图的邻接表

```
请输入图的顶点数和边数:
15 20
请输入图的顶点信息:
ABCDEFGHIJKLMNO
请输入图的边和权值:
0 1 1
0 4 1
1 2 1
2 12 1
2 5 1
3 0 1
3 7 1
4 9 1
5 1 1
6 3 1
6 14 1
7 11 1
8 4 1
8 11 1
9 6 1
10 7 1
11 10 1
12 4 1
13 6 1
14 1 1

图的邻接表为:
A->E(1)->B(1)->NULL
B->C(1)->NULL
C->F(1)->M(1)->NULL
D->H(1)->A(1)->NULL
E->J(1)->NULL
F->B(1)->NULL
G->O(1)->D(1)->NULL
H->L(1)->NULL
I->L(1)->E(1)->NULL
J->G(1)->NULL
K->H(1)->NULL
L->K(1)->NULL
M->E(1)->NULL
N->G(1)->NULL
O->B(1)->NULL
```

图 3 邻接表

### (4) 时间和空间复杂度分析

#### 1. 邻接矩阵

时间复杂度：邻接矩阵初始化  $O(n^2)$ 、读取顶点  $O(n)$ 、读取边  $O(e)$

$$T = O(n^2 + n + e) = O(n^2)$$

空间复杂度：由于邻接矩阵占用  $n^2$  个数据

$$S = O(n + e)$$

#### 2. 邻接表

时间复杂度：边表初始化、读取顶点  $O(n)$ 、读取边  $O(e)$

$$T = O(n + e)$$

空间复杂度：由于邻接表要存储顶点信息和边信息

$$S = O(n + e)$$

## 二、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。

### (1) 邻接矩阵转成邻接表

```
请选择操作：1.创建图的邻接矩阵;2.将邻接矩阵转邻接表;3.非递归深度优先搜索;4.递归深度优先搜索;5.广度优先搜索;0.退出
2
将邻接矩阵转换为邻接表：
A->E(1)->B(1)->NULL
B->C(1)->NULL
C->M(1)->F(1)->NULL
D->H(1)->A(1)->NULL
E->J(1)->NULL
F->B(1)->NULL
G->O(1)->D(1)->NULL
H->L(1)->NULL
I->L(1)->E(1)->NULL
J->G(1)->NULL
K->H(1)->NULL
L->K(1)->NULL
M->E(1)->NULL
N->G(1)->NULL
O->B(1)->NULL
```

图 4 邻接矩阵转成邻接表

### (2) 邻接表转成邻接矩阵

```
请选择操作：1.创建图的邻接表;2.将邻接表转邻接矩阵;3.非递归深度优先搜索;4.递归深度优先搜索;5.广度优先搜索;6.各个顶点的入度、出度、度;0.退出
2
将邻接表转换为邻接矩阵：
0 1 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 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 1 0 0
1 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 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 0
0 0 0 1 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
0 0 0 0 1 0 0 0 0 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 0 0 0 1 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 0 0 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 0 0 0 1 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
```

图 5 邻接表转成邻接矩阵

三、在上述两种存储结构上，分别实现无向图（或有向图）的深度优先搜索（递归和非递归）和广度优先搜索算法。

(1) 邻接矩阵

1. 非递归深度优先搜索

```
请选择操作：1.创建图的邻接矩阵;2.将邻接矩阵转邻接表;3.非递归深度优先搜索;4.递归深度优先搜索;5.广度优先搜索;0.退出
3
非递归深度优先序列：
1号森林：
A B C F M E J G D H L K O
2号森林：
I
3号森林：
N
非递归深度优先序号：
A:1
B:2
C:3
D:9
E:6
F:4
G:8
H:10
I:14
J:7
K:12
L:11
M:5
N:15
O:13
```

图 6 非递归深度优先搜索

2. 递归深度优先搜索

```
请选择操作：1.创建图的邻接矩阵;2.将邻接矩阵转邻接表;3.非递归深度优先搜索;4.递归深度优先搜索;5.广度优先搜索;0.退出
4
递归深度优先序列：
1号森林：
A B C F M E J G D H L K O
2号森林：
I
3号森林：
N
递归深度优先序号：
A:1
B:2
C:3
D:9
E:6
F:4
G:8
H:10
M:5
N:15
O:13
```

图 7 递归深度优先搜索

### 3. 广度优先搜索

```
请选择操作：1.创建图的邻接矩阵;2.将邻接矩阵转邻接表;3.非递归深度优先搜索;4.递归深度优先搜索;5.广度优先搜索;0.退出
5
广度优先序列：
1号森林：
A B E C J F M G D O H L K
2号森林：
I
3号森林：
N
广度优先序号：
A:1
B:2
C:4
D:9
E:3
F:6
G:8
H:11
I:14
J:5
K:13
L:12
M:7
N:15
O:10
```

图 8 广度优先搜索

### (2) 邻接表

### 1. 非递归深度优先搜索

```
请选择操作：1.创建图的邻接表;2.将邻接表转邻接矩阵;3.非递归深度优先搜索;4.递归深度优先搜索;5.广度优先搜索;6.各个顶点的入度、出度、度;0.退出
3
非递归深度优先序列：
1号森林：
A E J G O B C F M D H L K
2号森林：
I
3号森林：
N
非递归深度优先序号：
A:1
B:6
C:7
D:10
E:2
F:8
G:4
H:11
I:14
J:3
K:13
L:12
M:9
N:15
O:5
```

图 9 非递归深度优先搜索

## 2. 递归深度优先搜索

```
请选择操作：1.创建图的邻接表;2.将邻接表转邻接矩阵;3.非递归深度优先搜索;4.递归深度优先搜索;5.广度优先搜索;6.各个顶点的入度、出度、度;0.退出
4
递归深度优先序列：
1号森林：
A E J G O B C F M D H L K
2号森林：
I
3号森林：
N
递归深度优先序号：
A:1
B:6
C:7
D:10
E:2
F:8
G:4
H:11
I:14
J:3
K:13
L:12
M:9
N:15
O:5
```

图 10 递归深度优先搜索

## 3. 广度优先搜索

```
请选择操作：1.创建图的邻接表;2.将邻接表转邻接矩阵;3.非递归深度优先搜索;4.递归深度优先搜索;5.广度优先搜索;6.各个顶点的入度、出度、度;0.退出
5
广度优先序列：
1号森林：
A E B J C G F M O D H L K
2号森林：
I
3号森林：
N
广度优先序号：
A:1
F:7
G:6
H:11
I:14
J:4
K:13
L:12
M:8
N:15
O:9
```

图 11 广度优先搜索

## （3）时间和空间复杂度分析

### 1. 时间复杂度

深度优先搜索和广度优先搜索整个过程均要遍历所有结点，所以时间复杂度为：

邻接矩阵搜索：  $T = O(n^2)$

邻接表搜索：  $T = O(n + e)$

### 2. 空间复杂度

广度优先搜索和深度优先搜索都要使用数组，广度优先搜索还要使用队列。所以空间复杂度为：

邻接矩阵搜索：  $S = O(n)$

邻接表搜索：  $S = O(n)$

四、对于有向图，采用“邻接表”存储结构，设计和实现计算每个顶点入度、出度和度的算法，并分析其时间复杂度。

(1) 计算入度、出度、度

```
请选择操作：1.创建图的邻接表;2.将邻接表转邻接矩阵;3.非递归深度优先搜索;4.递归深度优先搜索;5.广度优先搜索;6.各个顶点的入度、出度、度;0.退出
6
各个顶点的入度、出度、度：
A的入度为:1,出度为:2,度为:3
B的入度为:3,出度为:1,度为:4
C的入度为:1,出度为:2,度为:3
D的入度为:1,出度为:2,度为:3
E的入度为:3,出度为:1,度为:4
F的入度为:1,出度为:1,度为:2
G的入度为:2,出度为:2,度为:4
H的入度为:2,出度为:1,度为:3
I的入度为:0,出度为:2,度为:2
J的入度为:1,出度为:1,度为:2
K的入度为:1,出度为:1,度为:2
L的入度为:2,出度为:1,度为:3
M的入度为:1,出度为:1,度为:2
N的入度为:0,出度为:1,度为:1
O的入度为:1,出度为:1,度为:2
```

图 12 计算入度、出度、度

(2) 时间复杂度分析

循环遍历了图中的所有顶点，每个顶点对应的循环内部会遍历该顶点的邻接表，对于每个邻接表中的边，会对相应的入度、出度以及度进行计数。

因此时间复杂度是  $T = O(n + e)$