数据结构作业

12章

1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 顶点 | 1 | 2 | 3 | 4 | 5 |
| 出度 | 2 | 0 | 2 | 1 | 1 |
| 入度 | 0 | 3 | 1 | 1 | 1 |
| 邻接于顶点2的点集 | 没有 | | | | |
| 邻接至顶点1的点集 | 没有 | | | | |
| 关联于顶点3的边集 | (3,5),(3,2) | | | | |
| 关联至顶点4的边集 | (1,4) | | | | |
| 有向环路 | 图中没有环路 | | | | |

a)

b)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 顶点 | 1 | 2 | 3 | 4 | 5 |
| 出度 | 0 | 2 | 1 | 3 | 1 |
| 入度 | 2 | 1 | 2 | 0 | 2 |
| 邻接于顶点2的点集 | 1,3 | | | | |
| 邻接至顶点1的点集 | 2,4 | | | | |
| 关联于顶点3的边集 | (3,5) | | | | |
| 关联至顶点4的边集 | 没有 | | | | |
| 有向环路 | ,长度3 | | | | |

c)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 顶点 | 1 | 2 | 3 | 4 | 5 |
| 出度 | 2 | 2 | 1 | 0 | 2 |
| 入度 | 1 | 1 | 1 | 3 | 1 |
| 邻接于顶点2的点集 | 3,5 | | | | |
| 邻接至顶点1的点集 | 5 | | | | |
| 关联于顶点3的边集 | (3,4) | | | | |
| 关联至顶点4的边集 | (1,4)(5,4) | | | | |
| 有向环路 | ,长度3 | | | | |

10. 图12—5 假设翻译人员从上到下编号为1，2，3，4，语言从上到下编号为5，6，7，8，9，则

邻接矩阵表示：

邻接压缩表：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | 7 | 5 | 6 | 7 | 9 | 5 | 7 | 1 | 2 | 3 | 2 | 1 | 4 | 2 | 23 |

h[1] h[2] h[3] h[4] h[5] h[6] h[7] h[8] h[9]

h=[0,2,5,6,8,11,12,14,15,16]

邻接链表：

|  |  |  |
| --- | --- | --- |
| [1] | 5 | 7 |
| [2] | 5 | 8 |
| [3] | 9 |
| [4] | 5 | 7 |
| [5] | 1 | 2 | 4 |
| [6] | 2 |
| [7] | 1 | 4 |
| [8] | 2 |
| [9] | 3 |

图12-8-a 1 2 3 4 5

邻接矩阵表示：

邻接压缩表：

h[1] h[2],h[3] h[4] h[5] h6[]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2 | 4 | 2 | 5 | 3 | 2 |

h=[0,2,2,4,5,6]

邻接链表：

|  |  |  |
| --- | --- | --- |
| [1] | 2 | 4 |
| [2] |
| [3] | 2 | 5 |
| [4] | 3 |
| [5] | 2 |

13. //将下三角矩阵按列映射

void restore (int i,int j,int e) {

if(i<j) swap(i,j);

else if(i==j) return ;

if(j==1) a[i-1]=e;

else {

int k=((n)\*(n+1)-(n-j+1)\*(n-j))/2;//计算第j列钱面的元素数

a[k+i-j]=e;

}

}

////将下三角矩阵按列映射

int retrieve(int i,int j,int e ){

if(i<j) swap(i,j);

else if(i==j) return e=0;

if(j==1) e=a[i-1];

else {

int k=((n)\*(n+1)-(n-j+1)\*(n-j))/2;//计算第j列钱面的元素数

e=a[k+i-j];

}

return e;

}

15. //确定点i出度，复杂度o(n)

int get\_outdegree(int i){

int cnt=0;

i--;

for(int j=0;j<n;++j){

if(a[i][j]) cnt++;

}

return cnt;

}

//确定点i出入度，复杂度o(n)

int get\_indegree(int i){

int cnt=0;

i--;

for(int j=0;j<n;++j){

if(a[j][i]) cnt++;

}

return cnt;

}

//确定图中边的数目，复杂度o()

int edges\_number(){

int cnt=0;

for(int i=0;i<n;++i){

for(int j=0;j<n;++j){

if(a[i][j]) cnt++;

}

}

return (cnt/2);

}

23.

图12-1-a

[1]

[2]

[3]

[4]

图12-1-b

[1]

[2]

[3]

[4]

[5]

[6]

[7]

45.

1)从顶点1开始的宽度优先生成树 2) 从顶点3开始的宽度优先生成树

1

1

3

2

3

2

4

4

3) 从顶点1开始的深度优先生成树 4)从顶点3开始的宽度优先生成树

1

1

2

3

2

3

4

4

48.

#define M 5010 //题目中可能的最大点数

int STACK[M],top=0; //Tarjan 算法中的栈

bool InStack[M]; //检查是否在栈中

int DFN[M]; //深度优先搜索访问次序

int Low[M]; //能追溯到的最早的次序

int ComponentNumber=0; //有向图强连通分量个数

int Index=0; //索引号

vector <int> Edge[M]; //邻接表表示

void Tarjan(int i)

{

int j;

DFN[i]=Low[i]=Index++;

InStack[i]=true;

STACK[++top]=i;

for (int e=0;e<Edge[i].size();e++)

{

j=Edge[i][e];

if (DFN[j]==-1)

{

Tarjan(j);

Low[i]=min(Low[i],Low[j]);

}

else if (InStack[j])

Low[i]=min(Low[i],DFN[j]);

}

if (DFN[i]==Low[i])

{

int cnt=0;//记录该钱联通分量的点数

do

{

j=STACK[top--];

InStack[j]=false;

}

while (j!=i);

if(cnt>1) ComponentNumber++;

}

}

void solve(int N) //N是此图中点的个数

{

memset(STACK,-1,sizeof(STACK));

memset(InStack,0,sizeof(InStack));

memset(DFN,-1,sizeof(DFN));

memset(Low,-1,sizeof(Low));

for(int i=0;i<N;i++)

if(DFN[i]==-1)

Tarjan(i);

}

template<class T>

bool Network::Cycle(){

slove();

if(ComponentNumber>=1) return 1;

else return 0;

}

13章

习题1

设矩阵从上到下一次的点依次为1,2,3,4

4

2

3

1

如果要使此图成为完全图，还需要补充7条边

习题2

Kruskal:

Prime:

习题3：

图的邻接链表表示：

[1]

[2]

[3]

[4]

[5]

[6]

图的其中一个最小生成树：

1

17

6

12

2

5

7

19

6

4

3

习题4：

宽度优先搜索树

2

1

3

4

6

5

深度优先搜索树：

1

2

3

4

6

5

习题5：

1)如表所示，点1到其各点的最短路径以及最短距离

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 顶点 | 2 | 3 | 4 | 5 |
| 最短路径 |  |  |  |  |
| 最短距离 | 10 | 18 | 15 | 17 |

2)如表所示，各顶点间的最短路径为

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 顶点 | 1 | 2 | 3 | 4 | 5 |
| 1 | ---------- | ---------- | ---------- | ---------- | ---------- |
| 2 |  | ---------- |  |  |  |
| 3 |  |  | ---------- |  |  |
| 4 |  |  |  | ---------- |  |
| 5 |  |  |  |  | ---------- |

14章