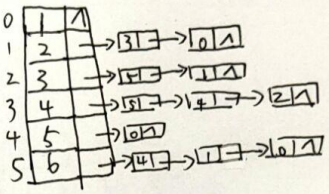
数据结构第六章

中国海洋大学
OCEAN UNIVERSITY OF CHINA

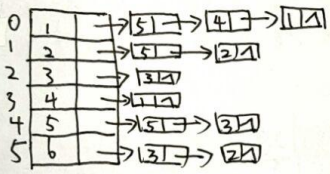
1、(1) ID(1)=3 OD(1)=0, ID(2)=2 OD(2)=2
ID(3)=1 OD(3)=2, ID(4)=1 OD(4)=3
ID(5)=2 OD(5)=1, ID(6)=2 OD(6)=3

(2)
$$\begin{bmatrix} 0&0&0&0&0&0 \\ 1&0&0&1&0&0 \\ 0&1&0&0&0&1 \\ 0&0&1&0&1&1 \\ 1&0&0&0&0&0 \\ 1&1&0&0&1&0 \end{bmatrix}$$

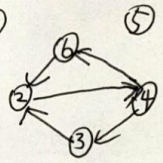
(3)



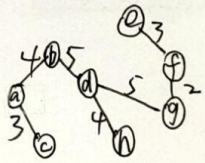
(4)



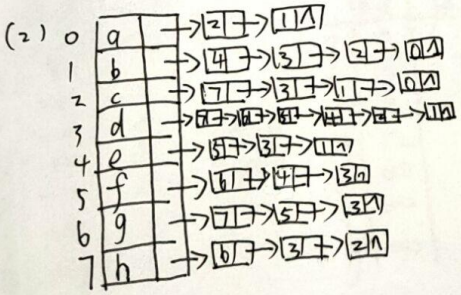
(5)

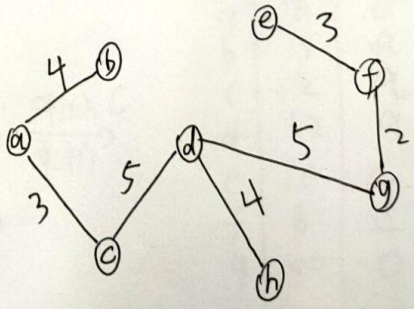


7、(1)
$$\begin{bmatrix} \infty&4&3&\infty&\infty&\infty&\infty&\infty \\ 4&\infty&5&5&9&\infty&\infty&\infty \\ 3&5&\infty&5&\infty&\infty&\infty&5 \\ \infty&5&5&\infty&7&6&5&4 \\ \infty&9&\infty&7&\infty&3&\infty&\infty \\ \infty&\infty&\infty&6&3&\infty&2&\infty \\ \infty&\infty&\infty&5&\infty&2&\infty&6 \\ \infty&\infty&5&4&\infty&\infty&6&\infty \end{bmatrix}$$



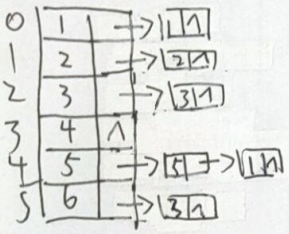
(2)





地址：青岛市松岭路238号 邮编：266100 电话：0532-66782730 传真：0532-66782799 网址：http://www.ouc.edu.cn

7.9 152364, 152634, 156234, 512364, 512634, 516234, 561234



Topological Sort 求得 561234

7.10

	α	A	B	C	D	E	F	G	H	I	J	K	W
Ve	0	1	6	17	3	34	4	3	13	1	31	22	44
Vl	0	20	24	26	19	34	8	3	13	7	31	22	44

	αA	αB	αD	αF	αG	αI	AC	BC	DC	AE	DJ	FE	FH	GW	GH	IH	CE	AC	GJ	HK	EW	JE	JW	KJ
e	0	0	0	0	0	0	1	6	3	3	3	4	4	3	3	1	17	13	13	13	34	31	31	22
l	19	18	16	4	0	6	20	24	19	26	25	23	8	23	13	7	26	24	17	15	34	31	32	22

关键路径 αGHKJEW

7.11

初始化

	Dist	Path
a	0	0
b	∞	0
c	∞	0
d	∞	0
e	∞	0
f	∞	0
g	∞	0

s中加入a，更新

	Dist	Path
a	0	0
b	15	a
c	2	a
d	12	a
e	∞	0
f	∞	0
g	∞	0

s中加入C，更新

	Dist	Path
a	0	0
b	15	a
c	2	a
d	12	a
e	10	c
f	6	c
g	∞	0

	Dist	Path
a	0	a
b	15	a
c	2	a
d	11	f
e	10	c
f	6	c
g	16	f

纳加入f 更新

	Dist	Path
a	0	a
b	15	a
c	2	a
d	11	f
e	10	c
f	6	c
g	16	f

纳加入e 更新

	Dist	Path
a	0	a
b	15	a
c	2	a
d	11	f
e	10	c
f	6	c
g	14	d

纳加入d 更新

	Dist	Path
a	0	a
b	15	a
c	2	a
d	11	f
e	10	c
f	6	c
g	14	d

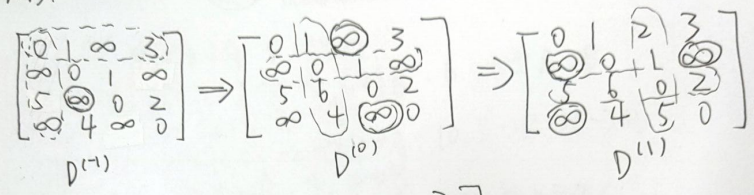
纳加入g 更新

	Dist	Path
a	0	a
b	15	a
c	2	a
d	11	f
e	10	c
f	6	c
g	14	d

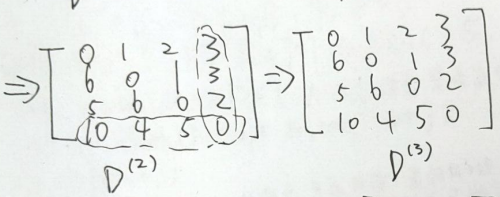
纳加入b 最终结果

a到b：15，a→b
a到c：2，a→c
a到d：11，a→c→f→d
a到e：10，a→c→e
a到f：6：a→c→f
a到g：14：a→c→f→d→g

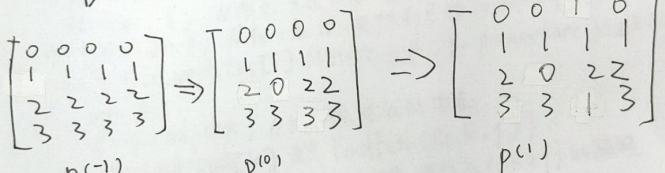
7.13.



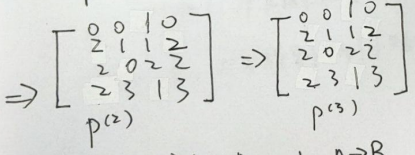
$D^{(-1)}$ $D^{(0)}$ $D^{(1)}$



$D^{(2)}$ $D^{(3)}$



$p^{(-1)}$ $p^{(0)}$ $p^{(1)}$



$p^{(2)}$ $p^{(3)}$

D[0][1]=1 ⇒A到B：1，A→B
D[0][2]=2，P[0][2]=1 ⇒A到C：2，A→B→C
D[0][3]=3，⇒A到D：3，A→D
D[1][0]=6，P[1][0]=2 ⇒B到A：6，B→C→A
D[1][2]=1，⇒B到C：1，B→C
D[1][3]=3，P[1][3]=2 ⇒B到D：3，B→C→D

D[2][0]=5 ⇒ C到A：5，C→A
D[2][1]=6，P[2][1]=0 ⇒ C到B：6，C→A→B
D[2][3]=2 ⇒ C到D：2，C→D
D[3][0]=10，P[3][0]=2 ⇒ D→A：10，D→B→C→A
D[3][1]=4，⇒ D到B：4，D→B
D[3][2]=5，P[3][2]=1 ⇒ D到C：5，D→B→C.

22、
```
int visited[MAXSIZE];//指示顶点是否在当前路径上
int FindPath(ALGraph G,int i,int j)
{ if(i==j)
    return 1; //顶点可达自身，递归的终止条件
  visited[i]=1; //值1表示i号顶点已收入路径中，
  for(p=G.vertices[i].firstarc;p;p=p->nextarc)//遍历以i为尾的弧
  { int k;
    k=p->adjvex; //k为弧指向的顶点
    if(!visited[k] && FindPath(G,k,j))
      return 1; //k顶点未被访问过，且存在从k到j的路径
  }
  return 0; //未找到i到j路径
}
```

23.
```
int visited[MAXSIZE]; //指示已走路径
int FindPath_BFS(ALGraph G, int i, int j)
{ InitQueue(Q);
  EnQueue(Q,i);
  while(!QueueEmpty(Q))
  { Deque(Q,u);   //出队列
    visited[u]=1; //表示u号顶点已被访问
    for(p=G.vertices[u].firstarc; p; p=p->nextarc)
    { //遍历以u为尾的弧,
      int k = p->adjvex;
      if(k==j)
        return 1; //k=j, u可达j
      if(!visited[k]) //u不可达j, k未被访问过
        EnQueue(Q,k); //则将k顶点入队
    }
  }
  return 0; //没有从i到j的路径
}
```

27.
```
int visited[MAXSIZE];
int FindLenPath(ALGraph G, int i, int j, int len)
{ if (len < 0)
    return 0; //不存在长度小于0的路径
  if (i==j && len==0)
    return 1; //顶点到自身距离为0，递归终止条件
  visited[i] = 1; // 访问i号顶点
  for (p=G.vertices[i].firstarc; p; p=p->nextarc)
  { int k = p->adjvex;
    if (!visited[k] && FindPath(G,k,j,len-1))
      return 1; //k未被访问且存在从k到j长度为len-1的简单路径
  }
  visited[i] = 0; //置该顶点为未访问，可在下种路径中被访问
}
```

34.
```
int SetNum(ALGraph G, int new[])
{ //拓扑排序,结果存入new中
  int indegree[MAXSIZE];
  FindIndegree(G, Indegree); //初始化各顶点入度
  InitStack(S);
  for(int i=0; i<G.vexnum; i++)
    if(!indegree[i]) //将所有入度为0的顶点入栈
      Push(S,i);
  count = 0; //初始化已重新编号顶点数为0
  while(!StackEmpty(S))
  { Pop(S,t);
    new[t] = ++count; //将出栈元素数作为出栈顶点的新编号
    for(p=G.vertices[t].firstarc; p; p=p->nextarc)
    { k=p->adjvex;
      if(!--indegree[k]) //k的度-1后入度为0
        Push(S,k);
    }
  }
  if(count==G.vexnum) //全部顶点已重新编号
    return 1;
  else
    return 0;
}
```

42

```
void Dijkstra(ALGraph G, int v0, Pathmatrix &P, ShortestPathTable &D)
{ for(i=0; i<G.vexnum; i++)
    D[i] = INFINITY;
  for(p=G.vertices[v0].firstarc; p; p=p->nextarc)
    D[p->adjvex] = *p->info; // 给D数组赋初值
  for(v=0; v<G.vexnum; v++)
  { final[v] = 0;
    for(w=0; w<G.vexnum; w++)
      P[v][w] = 0; // 设空路径
    if(D[v]<INFINITY)
    { P[v][v0]=1;
      P[v][v] = 1;
    }
  }
  D[v0] = 0;
  final[v0]=1;
  for(i=1; i<G.vexnum; i++)
  { min = INFINITY;
    for(w=0; w<G.vexnum; w++)
      if(!final[w])
        if(D[w]<min)
        { v=w;
          min = D[w];
        }
    final[v] = 1;
```

```
for(p=G.vertices[v].firstarc; p; p=p->nextarc)
{w=p->adjvex;
  if(!final[w]&&(min+(*p->info)<D[w])) //更短
    {D[w]=min+edglen(G,v,w);
     P[w]=P[v];
     P[w][w]=1; //构造最短路径
    }
  }
}
```