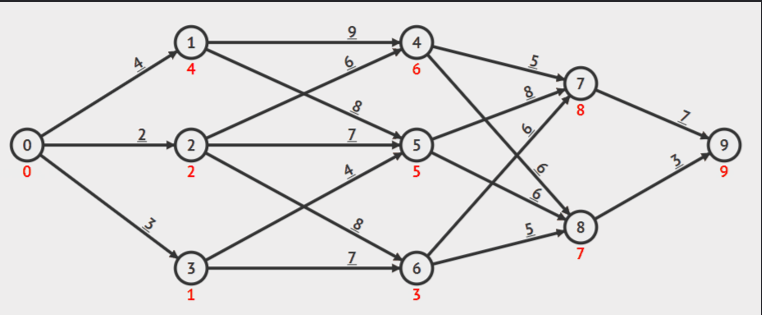
最短路径分析

求结点0~结点9的最短路径



**一.创建多段图**

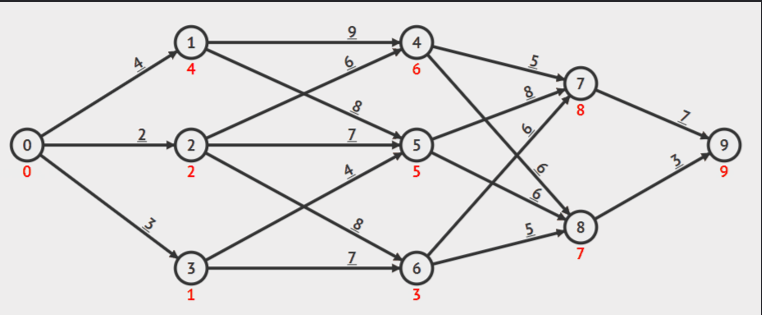
1.根据邻接矩阵创建多段图。

2.划分子集：因为多段图可以将顶点划分为k个互不相交的子集，所以将它划分为k段，每段都包含顶点，且每段的顶点互不相交。

3.为每个顶点编号，同一子集的顶点顺序无关紧要。源点s为0，终点t为n-1。

二. 证明满足最优性原理

设s，s1，s2，……，sp，t是s到t的一条最短路径，且s到s1的路径已经求出，则问题转为s1到t的最短路径，因此s1，s2，……，sp，t构成一条最短路径，如果不是，则设s1，r1，r2，……，rp，t是一条从s1到t的最短路径，则s，s1，r1，r2，……，rp，t是一条从s到t的最短路径且比s，s1，s2，……，sp，t短，因此矛盾，则满足最优性原理。



三、分析求解过程

自顶向下对多段图的边（u,v），用a(u,v)表示从u到v的边上的权值，将从终点t到源点s的最短路径记为d（t，s），则从终点9到源点0的最短路径d（9，0）由下式确定：

d（9,0）=min{a(9, 7)+d（7,0），a(9,8)+d（8,0）}

这是最后一个阶段的决策，它依赖于d（7,0）、d（8,0）的计算结果。

d（7,0）=min{a(7,4)+d（4,0），a(7,5)+d（5,0）, a(7,6)+d（6,0）}

d（8,0）=min{ a(8,4)+d（4,0），a(8,5)+d（5,0）, a(8,6)+d（6,0）}

这一阶段的决策又依赖于d（4,9）、d（5,9）和d（6,9）的计算结果。

d（4,0）=min{a(4,1)+d（1,0），a(4,2)+d（2,0）}

d（5,0）=min{a(5,1)+d（1,0），a(5,2)+d（2,0），a(5,3)+d（3,0）}

d（6,0）=min{a(6,2)+d（2,0），a(6,3)+d（3,0）}

这一阶段的决策依赖于d（1,0）、d（2,0）、d（3,0）的计算。

d（1,0）、d（2,0）和d（3,0）可以直接获得

d（1,0）=a(1,0)+d(0,0)=4（1→0）

d（2,0）=a(2,0)+d(0,0)=2（2→0）

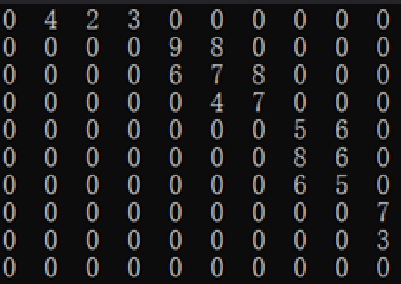
d（3,0）=a(3,0)+d(0,0)=3（3→0）

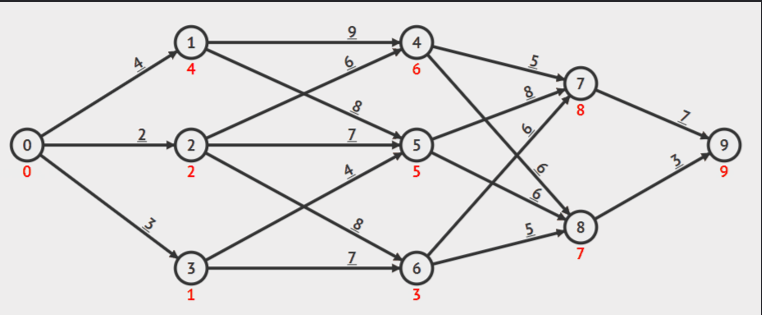
自底向上反向求解出最短的路径，得到：最短路径为0→3→5→8→9，长度为16。

算法实现

1. 图的邻接矩阵

0 1 2 3 4 5 6 7 8 9





设d和p两个数组

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| d | 0 | 4 | 2 | 3 | 8 | 7 | 10 | 13 | 13 | 16 |
| P | -1 | 0 | 0 | 0 | 2 | 3 | 2 | 4 | 5 | 8 |

d[i]:表示的是从0到i的最短路径

p[i]:表示i结点的前一个节点编号

初始：

d[0]=0

p[0]=-1

其它节点

d[i] = O;

p[i] = -1;

for (j = 1; j < n; j++)

{

for (i = j - 1; i >= 0; i--)

{

if (d[i]+a [i][j] < d[j])

{

d[j] = d[i]+a [i][j] ;

p[j] = i;

}

}

}

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| d | 0 | 4 | 2 | 3 | 8 | 7 | 10 | 13 | 13 | 16 |
| P | -1 | 0 | 0 | 0 | 2 | 3 | 2 | 4 | 5 | 8 |

结果输出

i=9,k=1

temp[9]

temp[0]=i

while (p [i] ！= -1)

{

Temp[k++]=p[i]

i = path[i];

}

For(i=k-1;i>=0;i++)

Printf(“%d->”,temp[i]);