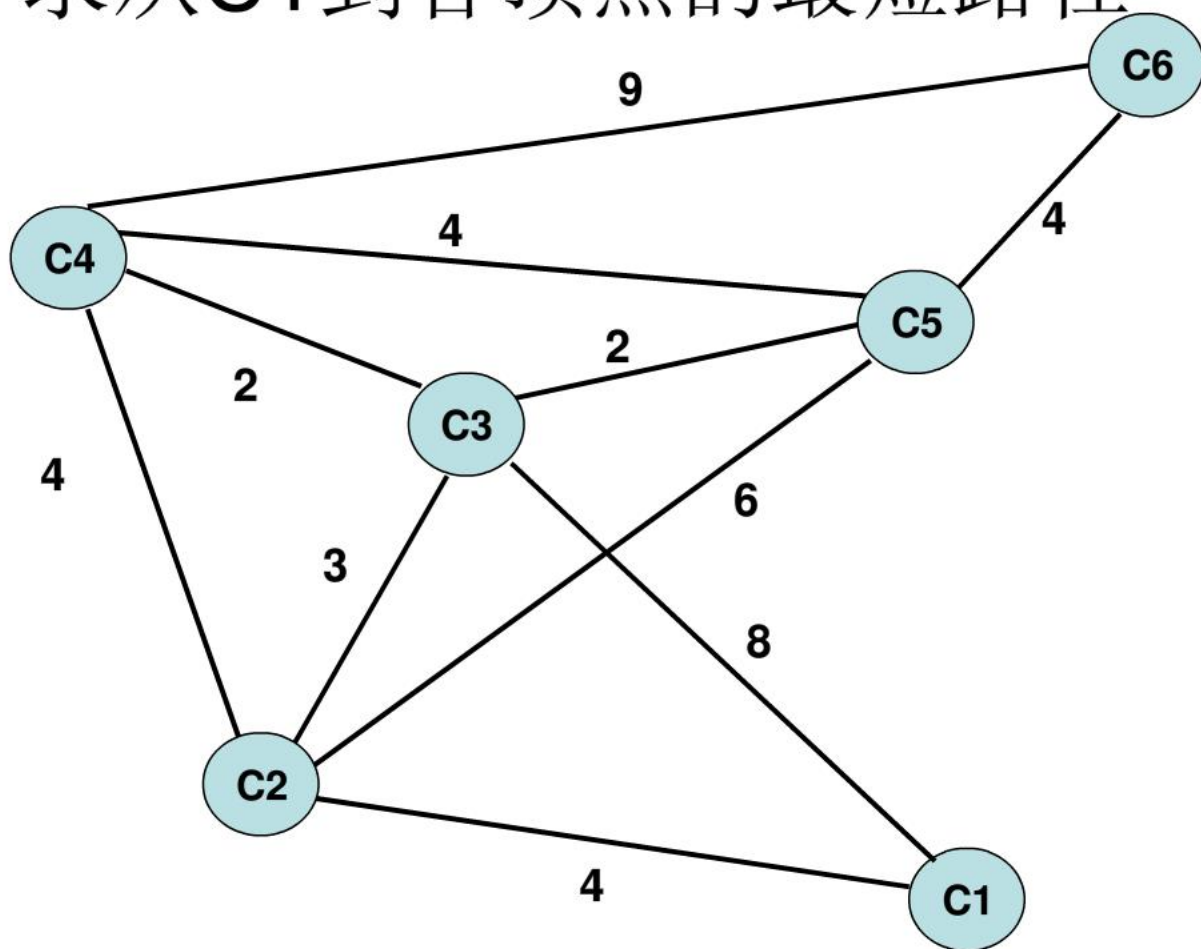


# 迪杰斯特拉（dijkstra）算法

- 用邻接矩阵**GA**来表示带权有向图；**s**为已找到从**v**出发的最短路径的终点的集合，它的初始状态为空集。那么，从**v**出发到图上其余各顶点**可能**达到的最短路径长度的初值为：  
 $\text{dist}[i] = \text{cost}[v_0, v_i] (v_i \in V)$
- 选择**v<sub>j</sub>**，使得 $\text{dist}[j] = \min\{\text{dist}[i] | (v_i \in V - s)\}$  **v<sub>j</sub>**就是当前求得的一条从**v<sub>0</sub>**出发的最短路径的终点。令**s**=**s** ∪ {**j**}
- 修改从**v**出发到集合**v-s**上任一顶点**v<sub>k</sub>**可达的最短路径长度。如果 $\text{dist}[j] + \text{cost}[j, k] < \text{dist}[k]$ , 则修改**dist[k]**为  
 $\text{dist}[k] = \text{dist}[j] + \text{cost}[j, k]$
- 重复上两步共**n-1**次，求得从到图上各顶点的最短路径是依路径长度递增的序列。

求从C1到各顶点的最短路径



```

□ Procedure dijkstra(GA,dist,path,i); {表示求Vi到图G中其余顶点的最短路径, GA为图G的邻接矩阵, dist和path为变量型参数, 其中path的基类型为集合}
□ begin
□   for j:=1 to n do   begin      {初始化}
□       if j<>i then s[j]:=0  else s[j]:=1;
□       dist[j]:=GA[i,j];
□       if dist[j]<maxint then path[j]:=[i]+[j]
□           else path[j]:=[ ];
□   end;
□   for k:=1 to n-2 do
□       begin
□         w:=maxint; m:=i;
□         for j:=1 to n do {求出第k个终点Vm}
□             if (s[j]=0) and (dist[j]<w) then begin m:=j;w:=dist[j];end;
□             if m<>i then s[m]:=1 else exit; {若条件成立, 则把Vm加入到s中, 否则
退出循环, 因为剩余的终点, 其最短路径长度均为maxint, 无需再计算下去}
□         for j:=1 to n do {对s[j]=0的更优元素作必要修改}
□             if (s[j]=0) and (dist[m]+GA(m,j)<dist[j])
□                 then begin
□                     dist[j]:=dist[m]+GA(m,j);
□                     path[j]:=path[m]+[j];
□                 end;
□         end;
□   end;
□ end;

```

初始时

	1	2	3	4	5	6
dist	0	4	8	maxint	maxint	maxint
path	C1	C1,C2	C1,C3			

第一次:选择m=2,则s=[c1,c2],计算比较dist[2]+GA[2,j]与dist[j]的大小(3<=j<=6)

	1	2	3	4	5	6
dist	0	4	7	8	10	maxint
path	C1	C1,C2	C1,c2,C3	C1,c2,c4	C1,c2,c5	

第二次:选择m=3,则s=[c1,c2,c3],计算比较dist[3]+GA[3,j]与dist[j]的大小(4<=j<=6)

	1	2	3	4	5	6
dist	0	4	7	8	9	maxint
path	C1	C1,C2	C1,c2,C3	C1,c2,c4	C1,c2,c3,c5	

第三次:选择m=4,则s=[c1,c2,c3,c4],计算比较dist[4]+GA[4,j]与dist[j]的大小(5<=j<=6)

	1	2	3	4	5	6
dist	0	4	7	8	9	17
path	C1	C1,C2	C1,c2,C3	C1,c2,c4	C1,c2,c3,c5	C1,c2,c4,c6

第四次:选择m=5,则s=[c1,c2,c3,c4,c5],计算比较dist[5]+GA[5,j]与dist[j]的大小(j=6)

	1	2	3	4	5	6
dist	0	4	7	8	9	13
path	C1	C1,C2	C1,c2,C3	C1,c2,c4	C1,c2,c3,c5	C1,c2,c3,c5,c6

# 迪杰斯特拉（dijkstra）算法

- For i:=1 to vtxnum do
- begin dist[i]:=cost[v0,i];
- if dist[i]<max then path[i]:=[v0]+[i]
- else path[i]:=[ ];
- end;
- S:=[v0];
- For k:=1 to vtxnum-1 do
- begin wm:=max; j:=v0;
- for i:=1 to vtxnum do
- if not (i in s) and (dist[i]<wm) then
- begin j:=i; wm:=dist[i] end;
- s:=s+[j];
- for i:=1 to vtxnum do
- if not (i in s) and (dist[j]+cost[j,i]<dist[i]) then
- begin dist[i]:=dist[j]+cost[j,i]; path[i]:=path[j]+[i] end
- end
- end