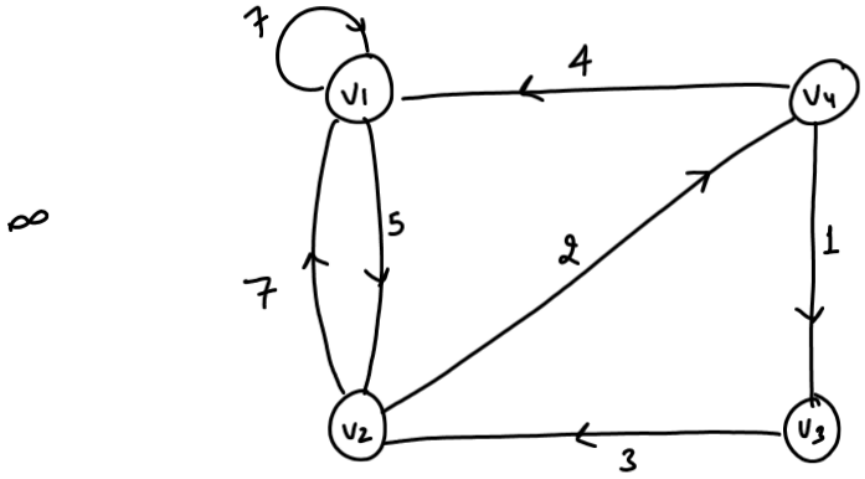
WARSHALL'S ALGORITHM FOR CALCULATING SHORTEST PATH MATRIX



∞

$$W = \text{(weight } M_k\text{)} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 7 & 5 & 0 & 0 \\ 7 & 0 & 0 & 2 \\ 0 & 3 & 0 & 0 \\ 4 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

$$M_0 \Rightarrow \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 7 & 5 & \infty & \infty \\ 7 & \infty & \infty & 2 \\ \infty & 3 & \infty & \infty \\ 4 & \infty & 1 & \infty \end{bmatrix} \end{array}$$

$$M_k[I][J] = Min\left(M_{k-1}[I][J],\ M_{k-1}[I][K] + M_{k-1}[K][J]\right)$$

$$M_1 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 7 & 5 & \infty & \infty \\ 7 & 12 & \infty & 2 \\ \infty & 3 & \infty & \infty \\ 4 & 9 & 1 & \infty \end{bmatrix} \end{array}$$

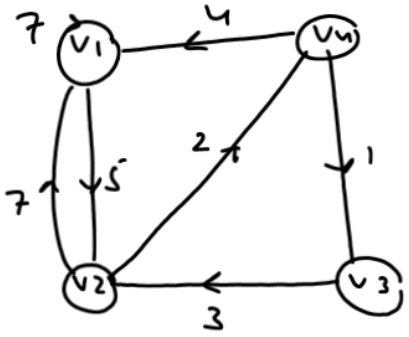
$M_1[4][2] = Min(M_0[4][2],\ M_0[4][1] + M_0[1][2])$

$= Min(\infty,\ 4+5)$

$$M_2 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 7 & 5 & \infty & 7 \\ 7 & 12 & \infty & 2 \\ 10 & 3 & \infty & 5 \\ 4 & 9 & 1 & 11 \end{bmatrix} \end{array}$$

$$M_3 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 7 & 5 & \infty & 7 \\ 7 & 12 & \infty & 2 \\ 10 & 3 & \infty & 5 \\ 4 & 4 & 1 & 6 \end{bmatrix} \end{array}$$

$$M_4 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 7 & 5 & 8 & 7 \\ 6 & 6 & 3 & 2 \\ 9 & 3 & 6 & 5 \\ 4 & 4 & 1 & 6 \end{bmatrix} \end{array}$$

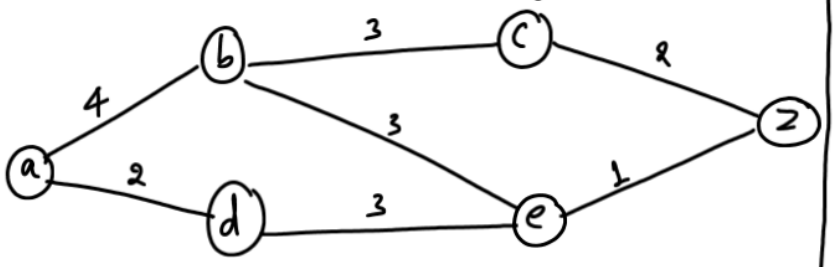



```
void shortest(int [][4],int [][4];);
int main()
{
  int weight[4][4],sp[4][4];
  int i,j;
  for(i=0;i<4;i++)
  {
    for(j=0;j<4;j++)
    {
      printf("Enter weight of path from v[%d] to v[%d] or 0 if no path present",i+1,j+1);
      scanf("%d",&weight[i][j]);
    }
  }
  shortest(weight,sp);
  printf("Shortest path is:\n");
  for(i=0;i<4;i++)
  {
    for(j=0;j<4;j++)
    {
      printf("%d ",sp[i][j]);
    }
  printf("\n");
 }
return 0;
}
```

```
void shortest(int w[4][4],int sp[4][4])
{
    int i,j,k;
    int INFINITY=100000;
   for(i=0;i<4;i++)
   {
       for(j=0;j<4;j++)
       {
           (w[i][j]!=0)?(sp[i][j]=w[i][j]):(sp[i][j]=INFINITY);
       }
  }

for(k=0;k<4;k++)
 {
    for(i=0;i<4;i++)
    {
       for(j=0;j<4;j++)
       {
           sp[i][j]=min(sp[i][j],sp[i][k]+sp[k][j]);
       }
    }
}
```

```
int min(int x,int y)
{
return (x<y?x:y);
}
```



p[a] = b, d
p[b] = c, e
p[c]
p[d] = e
p[e] = z
p[z]

$l(a) = 0,\ l(b) = \infty,\ l(c) = \infty,\ l(d) = \infty,\ l(e) = \infty,\ l(z) = \infty$

Step 1

Min = $l(a) = 0$

$l(b) = \text{Min}(l(b), w(a,b) + l(a))$
$= \text{Min}(\infty, 4 + 0) = 4$

$l(d) = \text{Min}(l(d), w(a,d) + l(a))$
$= \text{Min}(\infty, 2 + 0) = 2$

z → e → d → a

a → d → e → z

SP => G

Step 2 $l(b) = 4,\ l(d) = 2,\ l(c) = \infty,\ l(e) = \infty,\ l(z) = \infty$

$Min = l(d) = 2$

$l(e) = Min(\ l(e),\ w(d,e) + l(d))$

$= Min(\ \infty,\ 3+2) = 5$

Step 3 $l(b) = 4,\ l(e) = 5,\ l(c) = \infty,\ l(z) = \infty$

$Min = l(b) = 4$

$l(c) = Min(l(c),\ w(b,c) + l(b))$

$= Min(\infty,\ 3+4) = 7$

$l(e) = Min(l(e), w(b,e) + l(b))$

$= Min(5,\ 3+4) = 5$

Step 4 $l(c) = 7,\ l(e) = 5,\ l(z) = \infty$

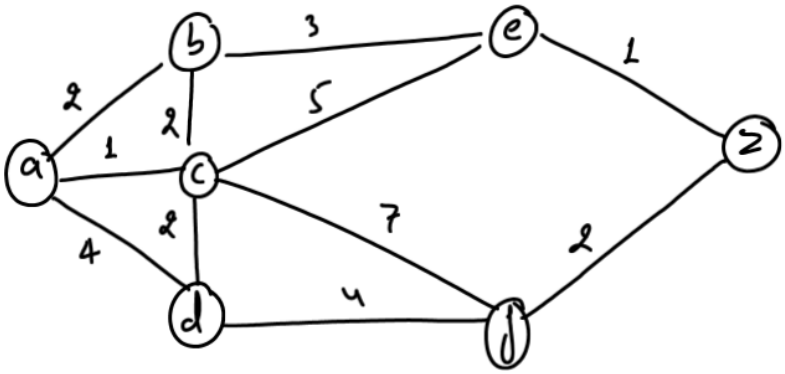
$Min = l(e) = 5$

$l(z) = Min(\ l(z),\ w(e,z) + l(e))$

$= Min(\ \infty,\ 1+5) = 6$

Step 5 $l(c) = 7,\ l(z) = 6$

$Min = l(z) = 6$



Find out the SHORTEST PATH from 'a' to 'z' as well as its length