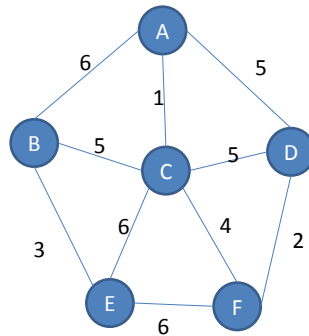


一、输入为无向网时候，输出为邻接矩阵,深度优先遍历和广度优先遍历，prim 最小生成树：

无向网



输入：

请输入图的类型(0-有向图，1-有向网，2-无向图，3-无向网)

3

G.vexnum :6

G.arcnum :10

请输入各顶点的信息(char):

G.vexs[0] : A

G.vexs[1] : B

G.vexs[2] : C

G.vexs[3] : D

G.vexs[4] : E

G.vexs[5] : F

请输入每条边依附的顶点及权值

input arc 1:

arc from v1 (char) : A

arc to v2 (char) : B

weight (int) : 6

input arc 2:

arc from v1 (char) : B

arc to v2 (char) : E

weight (int) : 3

input arc 3:

arc from v1 (char) : E

arc to v2 (char) : F

weight (int) : 6

input arc 4:

arc from v1 (char) : F

arc to v2 (char) : D

weight (int) : 2

input arc 5:

arc from v1 (char) : D

arc to v2 (char) : A

weight (int) : 5

input arc 6:

arc from v1 (char) : C

arc to v2 (char) : A

weight (int) : 1

input arc 7:

arc from v1 (char) : C

arc to v2 (char) : B  
weight (int) : 5  
input arc 8:  
arc from v1 (char) : C  
arc to v2 (char) : E  
weight (int) : 6  
input arc 9:  
arc from v1 (char) : C  
arc to v2 (char) : F  
weight (int) : 4  
input arc 10:  
arc from v1 (char) : C  
arc to v2 (char) : D  
weight (int) : 5

输出:

无向图 G

The MGraph G is:

A B C D E F

A |  $\infty$  6 1 5  $\infty$   $\infty$

B | 6  $\infty$  5  $\infty$  3  $\infty$

C | 1 5  $\infty$  5 6 4

D | 5  $\infty$  5  $\infty$   $\infty$  2

E |  $\infty$  3 6  $\infty$   $\infty$  6

F |  $\infty$   $\infty$  4 2 6  $\infty$

深度优先搜索的结果:

A B C D F E

广度优先搜索的结果:

A B C D E F

普里姆算法计算最小生成树,请输入起始的顶点 u:

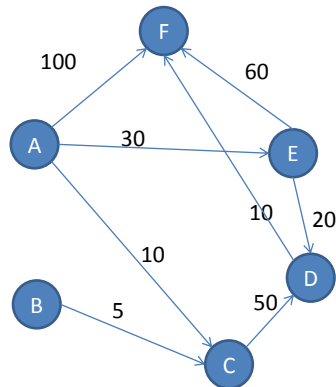
A

生成树的各边为:

A--C C--F F--D C--B B--E 请按任意键继续...

二、输入为有向网时候，输出为邻接矩阵,深度优先遍历和广度优先遍历，最短路径（两个算法）：

有向网



输入：

请输入图的类型(0-有向图，1-有向网，2-无向图，3-无向网)

1

G.vexnum :6

G.arcnum :8

请输入各顶点的信息(char):

G.vexs[0] : A

G.vexs[1] : B

G.vexs[2] : C

G.vexs[3] : D

G.vexs[4] : E

G.vexs[5] : F

请输入每条边依附的顶点及权值

input arc 1:

arc from v1 (char) : A

arc to v2 (char) : F

w (int) : 100

input arc 2:

arc from v1 (char) : A

arc to v2 (char) : E

w (int) : 30

input arc 3:

arc from v1 (char) : A

arc to v2 (char) : C

w (int) : 10

input arc 4:

arc from v1 (char) : B

arc to v2 (char) : C

w (int) : 5

input arc 5:

arc from v1 (char) : C

arc to v2 (char) : D

w (int) : 50

input arc 6:

arc from v1 (char) : D

arc to v2 (char) : F

w (int) : 10

input arc 7:

arc from v1 (char) : E

arc to v2 (char) : D  
w (int) : 20  
input arc 8:  
arc from v1 (char) : E  
arc to v2 (char) : F  
w (int) : 60

输出:

有向网 G

The MGraph G is:

	A	B	C	D	E	F
A	∞	∞	10	∞	30	100
B	∞	∞	5	∞	∞	∞
C	∞	∞	∞	50	∞	∞
D	∞	∞	∞	∞	∞	10
E	∞	∞	∞	20	∞	60
F	∞	∞	∞	∞	∞	∞

深度优先搜索的结果:

A C D F E B

广度优先搜索的结果:

A C E F D B

迪杰斯特拉算法计算最短路径,请输入起始的顶点 u:

A

计算结果如下:

最短距离:

D[A-A]= 0 D[A-B]= 2147483647 D[A-C]= 10 D[A-D]= 50 D[A-E]= 30 D[A-F]= 60

路径 (若 P[v][w]为 TRUE, 则 w 是从 v0 到 v 当前求得最短路径上的顶点。):

	A	B	C	D	E	F
A	0	0	0	0	0	0
B	0	0	0	0	0	0
C	1	0	1	0	0	0
D	1	0	0	1	1	0
E	1	0	0	0	1	0
F	1	0	0	1	1	1

采用 FLOYD 算法计算各个顶点间的最短路径

计算结果如下

各个顶点间的最短距离为:

	A	B	C	D	E	F
A	^ 10	50	30	60		
B	^ 5	55	^ 65			
C	^ ^ 50	^ 60				
D	^ ^ ^ 10					
E	^ ^ ^ 20	^ 30				
F	^ ^ ^ ^ 10					

各个顶点的最短路径为:

A 到其他顶点的最短路径为:

	A	B	C	D	E	F
A	0	0	0	0	0	0
B	0	0	0	0	0	0
C	1	0	1	0	0	0
D	1	0	0	1	1	0
E	1	0	0	0	1	0
F	1	0	0	1	1	1

B 到其他顶点的最短路径为:

	A	B	C	D	E	F
A	0	0	0	0	0	0
B	0	0	0	0	0	0
C	0	1	1	0	0	0
D	0	1	1	1	0	0
E	0	0	0	0	0	0

F011101

C 到其他顶点的最短路径为:

A B C D E F

A000000

B000000

C000000

D001100

E000000

F001101

D 到其他顶点的最短路径为:

A B C D E F

A000000

B000000

C000000

D000000

E000000

F000101

E 到其他顶点的最短路径为:

A B C D E F

A000000

B000000

C000000

D000110

E000000

F000111

F 到其他顶点的最短路径为:

A B C D E F

A000000

B000000

C000000

D000000

E000000

F000000

请按任意键继续...