

2.1. (1) 入/出度

	1	2	3	4	5	6
入度	3	2	1	1	2	2
出度	0	2	2	3	1	3

(2) 邻接矩阵

	1	2	3	4	5	6
1	0	0	0	0	0	0
2	1	0	0	0	0	0
3	0	1	0	0	0	1
4	0	0	1	0	1	1
5	1	0	0	0	0	0
6	1	1	0	0	1	0

(3) 邻接表

	V_1	V_2	V_3	V_4	V_5	V_6
0	\wedge					
1	$\rightarrow 3$	$\rightarrow 0 \wedge$				
2	$\rightarrow 5$	$\rightarrow 1 \wedge$				
3	$\rightarrow 5$	$\rightarrow 4$	$\rightarrow 2 \wedge$			
4	$\rightarrow 0 \wedge$					
5	$\rightarrow 4$	$\rightarrow 1$	$\rightarrow 0 \wedge$			

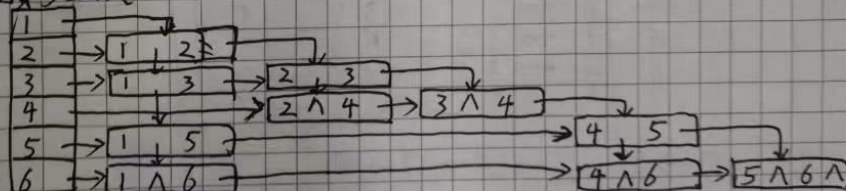
(4) 逆邻接表

	V_1	V_2	V_3	V_4	V_5	V_6
0	$\rightarrow 5$	$\rightarrow 4$	$\rightarrow 1 \wedge$			
1	$\rightarrow 5$	$\rightarrow 2 \wedge$				
2	$\rightarrow 3 \wedge$					
3	$\rightarrow 1 \wedge$					
4	$\rightarrow 5$	$\rightarrow 3 \wedge$				
5	$\rightarrow 3$	$\rightarrow 2 \wedge$				

(5) 有三个连通分量

1, 5, 2346

7.3 邻接多重表:

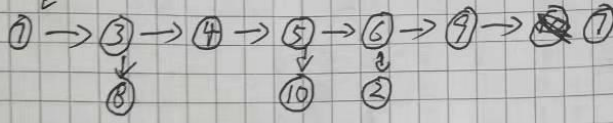


深度优先搜索: 1 5 6 4 3 2

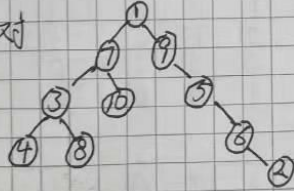
广度优先搜索: 1 5 6 3 2 4

7.4 广度优先搜索: ALFCBIMDEIGKH

7.5 ① 深度优先生成树



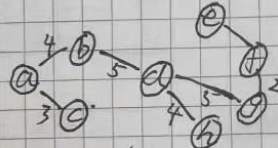
广度优先生成树



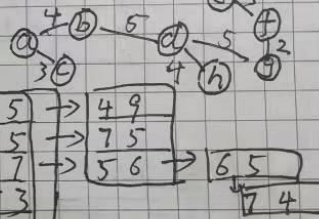
7.7 邻接矩阵

	a	b	c	d	e	f	g	h
a	0	1	1	0	0	0	0	0
b	1	0	1	1	0	0	0	0
c	1	1	0	1	0	0	0	1
d	0	1	1	0	1	1	1	1
e	0	1	0	1	0	1	0	0
f	0	0	0	1	1	0	1	0
g	0	0	0	1	0	1	0	1
h	0	0	1	1	0	0	1	0

Prim 生成树:



Kruskal 生成树



邻接表

	a	b	c	d	e	f	g	h
a	→ 1 4 → 2 3 1							
b	→ 0 4 → 2 5 → 3 5 → 4 9							
c	→ 0 3 → 1 5 → 3 5 → 7 5							
d	→ 1 5 → 2 5 → 4 7 → 5 6							
e	→ 1 9 → 3 7 → 5 3							
f	→ 3 6 → 4 3 → 6 2							
g	→ 3 5 → 5 2 → 7 6							
h	→ 2 5 → 3 4 → 6 6							

7.9 ① → ② → ③ → ⑤ → ⑥ → ④

	VE	VL
α	1	0
A	2	1
B	3	6
D	4	3
F	5	4
G	6	3
I	7	1
C	8	17
H	9	13
E	10	34
J	11	31
K	12	22
W	13	44

关键路径: α → G → H → J → W

α → G → H → J → E

α → G → H → K

7.11

		a	b	c	d	e	f	g	
①	dis	0	x	x	x	x	x	x	$x = inf$
	vis	0	0	0	0	0	0	0	dijkstra 堆优化
②	dis	0	15	2	12	x	x	x	
	vis	1	0	0	0	0	0	0	
③	dis	0	15	2	12	10	6	x	
	vis	1	0	1	0	0	0	0	
④	dis	0	15	2	11	10	6	16	
	vis	1	0	1	0	0	1	0	
⑤	dis	0	15	2	11	10	6	16	
	vis	1	0	1	0	1	1	0	
⑥	dis	0	15	2	11	10	6	14	
	vis	1	0	1	1	1	1	0	
⑦	dis	0	15	2	11	10	6	14	
	vis	1	1	1	1	1	1	1	

7.13.

	a	b	c	d	e	f	g	
a	x	15	2	11	10	6	14	$x = inf$
b	x	19	x	x	6	x	15	\Rightarrow Floyd 求最短路径
c	x	16	x	9	8	4	12	起点 终点
d	x	7	x	x	13	x	3	
e	x	13	x	x	19	x	9	
f	x	12	x	5	18	x	8	
g	x	4	x	x	10	x	19	

7.14

```

1 //作业7.14 By2020101603 阮炜霖
2 #include<bits/stdc++.h>
3 #define inf 0x3f3f3f3f
4 #define int long long
5 using namespace std;
6 const int N=2e5+7;
7 const int mod=1e9+7;
8
9 //int read(){ int x=0,f=1;char ch=getchar();while(ch<'0' || ch>'9'){if(ch=='-') f=f*-1;c
10 struct Edge{
11     int v,w;
12 };
13 vector<Edge>G[N];
14 int n,m,u,v,w;
15
16 signed main(){
17     // ios::sync_with_stdio(0);// cin.tie(0);cout.tie(0);
18     // freopen("in.cpp","r",stdin);freopen("out.cpp","w",stdout);
19     cin>>n>>m; //输入顶点和弧的数目
20     for(int i=1;i<=m;i++){ //输入各弧的信息
21         cin>>u>>v>>w;
22         G[u].push_back((Edge){v,w});
23     }
24     cout<<"邻接表: 结点表示为(to,dis->)\n";
25     for(int i=1;i<=n;i++){
26         cout<<i<<": ";
27         for(int j=0;j<G[i].size();j++){
28             cout<<G[i][j].v<<" "<<G[i][j].w<<"->";
29         }
30         cout<<"\n";
31     }
32     return 0;
33 }
34

```