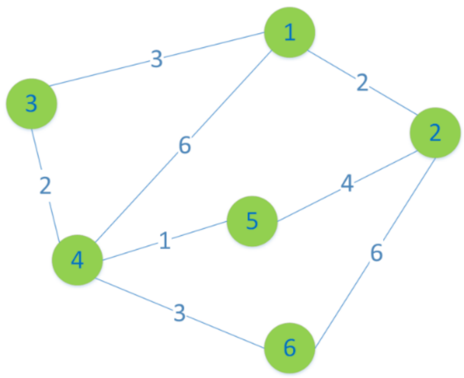
运筹学第十二次作业参考答案

1. （1）用 Floyd 算法计算所有点间的最短路

 （2）用 Bellman-Ford-Moore 算法和 Dijkstra 算法求解 v1 点到 v6 点的最短路



（1）

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

加入 1

	1	2	3	4	5	6
1	0	2	3	6	∞	∞
2	2	0	5	8	4	6
3	3	5	0	2	∞	∞
4	6	8	2	0	1	3
5	∞	4	∞	1	0	∞
6	∞	6	∞	3	∞	0

加入 2

	1	2	3	4	5	6
1	0	2	3	6	6	8
2	2	0	5	8	4	6
3	3	5	0	2	9	11
4	6	8	2	0	1	3
5	6	4	9	1	0	10
6	8	6	11	3	10	0

加入 3

	1	2	3	4	5	6
1	0	2	3	5	6	8
2	2	0	5	7	4	6
3	3	5	0	2	9	11
4	5	7	2	0	1	3
5	6	4	9	1	0	10
6	8	6	11	3	10	0

加入 4

	1	2	3	4	5	6
1	0	2	3	5	6	8
2	2	0	5	7	4	6
3	3	5	0	2	**3**	**5**
4	5	7	2	0	1	3
5	6	4	**3**	1	0	**4**
6	8	6	**5**	3	**4**	0

加入 5

	1	2	3	4	5	6
1	0	2	3	5	6	8
2	2	0	5	**5**	4	6
3	3	5	0	2	3	5
4	5	**5**	2	0	1	3
5	6	4	3	1	0	4
6	8	6	5	3	4	0

加入 6

	1	2	3	4	5	6
1	0	2	3	5	6	8
2	2	0	5	5	4	6
3	3	5	0	2	3	5
4	5	5	2	0	1	3
5	6	4	3	1	0	4
6	8	6	5	3	4	0

（2）

Bellman-Ford-Moore：

	1	2	3	4	5	6
$k=1$	0	2	3	6		
$k=2$	0	2	3	5	6	8
$k=3$	0	2	3	5	6	8

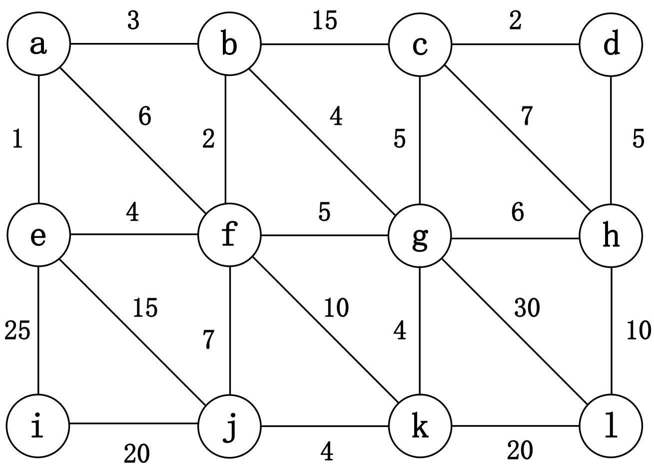
$f_3(v_i)=f_2(v_i), \forall i$，停止

Dijkstra：

	1	2	3	4	5	6	加入
$k=1$	0	**2**	3	6			v_1v_2
$k=2$	0	2	**3**	6	6	8	v_1v_3
$k=3$	0	2	3	**5**	6	8	v_3v_4
$k=4$	0	2	3	5	**6**	8	v_4v_5或v_2v_5
$k=5$	0	2	3	5	6	**8**	v_4v_6或v_2v_6

v_1到v_6的最短距离为 8，路径为 1->2->6 或 1->3->4->6

2. 分别用 Kruskal 算法和 Prim 算法求下图的最小生成树。



解：

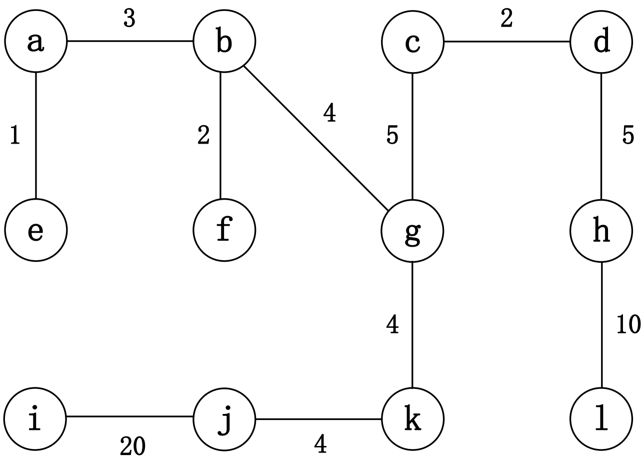
Kruskal 算法：

(a, e) = 1, (c, d) = 2, (b, f) = 2, (a, b) = 3, (e, f) = 4, (b, g) = 4 (j, k) = 4, (g, k) = 4, (f, g) = 5, (c, g) = 5, (d, h) = 5, (a, f) = 6, (g, h) = 6, (c, h) = 7, (f, j) = 7, (f, k) = 10, (h, l) = 10, (b, c) = 15, (e, j) = 15, (i, j) = 20, (k, l) = 20 (e, i) = 25, (g, l) = 30

从小到大顺序选择不构成回路的边形成支撑树

Prim 算法：

a	b	c	d	e	f	g	h	i	j	k	l	加入
0	3	∞	∞	**1**	6	∞	∞	∞	∞	∞	∞	ae
0	**3**	∞	∞	1	4	∞	∞	25	15	∞	∞	ab
0	3	15	∞	1	**2**	4	∞	25	15	∞	∞	bf
0	3	15	∞	1	2	**4**	6	25	7	10	∞	bg
0	3	5	∞	1	2	4	6	25	7	**4**	30	gk
0	3	5	∞	1	2	4	6	25	**4**	4	20	kj
0	3	**5**	∞	1	2	4	6	20	4	4	20	cg
0	3	5	**2**	1	2	4	6	20	4	4	20	cd
0	3	5	2	1	2	4	**5**	20	4	4	20	dh
0	3	5	2	1	2	4	5	20	4	4	**10**	hl
0	3	5	2	1	2	4	5	**20**	4	4	10	ij



3. 修改下图 Floyd 算法，使得 k 循环为最内层循环。使用修改后的算法重新计算第 1 题的（1），并解释修改后的算法为什么可行或为什么不可行。

```
int d[MAXN][MAXN]; // d[u][v] 表示从u -> v的权值 不存在的时候为0
int V; // 顶点个数

void Floyd()
{
    for(int k = 0;k < V;k++)
        for(int i = 0;i < V;i++)
            for(int j = 0;j < V;j++)
                d[i][j] = min(d[i][j],d[i][k]+d[k][j]);
}
```

修改后重新计算第 1 题的（1），结果不变。

但是这是因为图中任意两点间最短通路数≤ 2，对一般的图，d[i][j]计算时 k 相关的路径信息可能还未更新，从而导致计算错误。