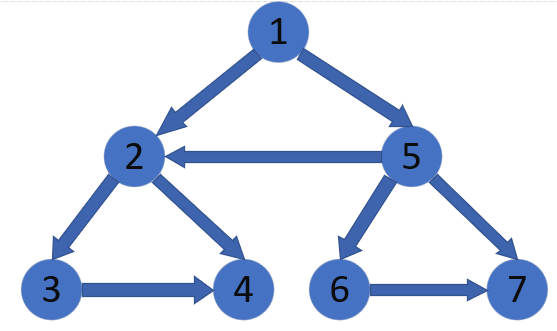
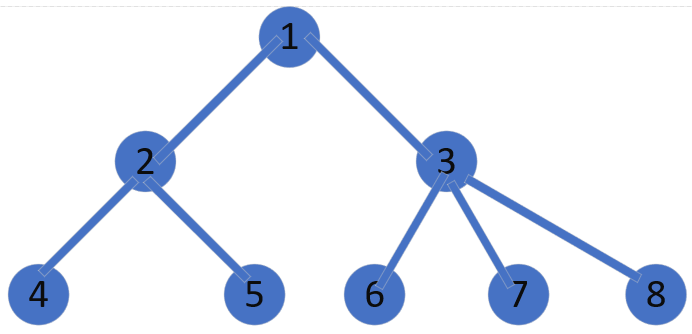
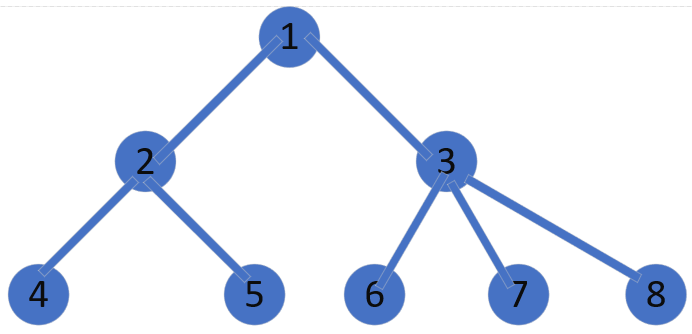
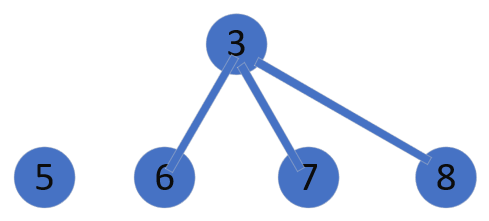
6.10对下列图分别执行dfs(1)和dfs(5),写出遍历序列，并构造出相应的dfs生成树。

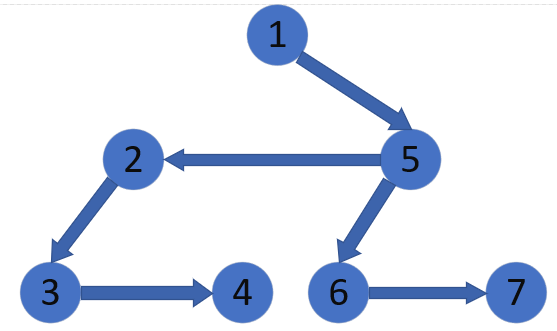
dfs(1):1 2 4 5 3 6 7 8



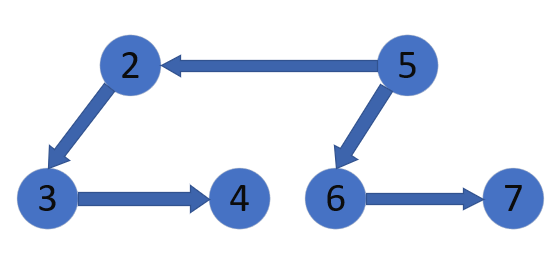
dfs(5):5 3 6 7 8



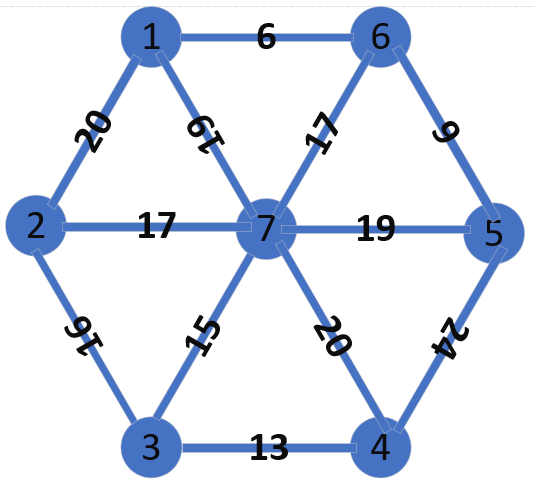
dfs(1):1 5 2 3 4 6 7



dfs(5): 5 2 3 4 6 7



6.23分别用prim算法和Kruskal算法求解下图的最小生成树，标注中间的求解过程



Prim

初始化S={ },U={1,2,3,4,5,6,7},WE={ },TE={ }.

第一轮选择S={1},U={2,3,4,5,6,7},WE={(1,2),(1,7),(1,6)}选择权值最小的(1,6)

S={1},U={2,3,4,5,6,7},WE={(1,2),(1,7)},TE={(1,6)}

第二轮

S={1,6},U={2,3,4,5,7},WE={(1,2),(1,7)(6,7),(6,5)}选择权值最小的(6,5)

S={1,6},U={2,3,4,5,7},WE={(1,2),(1,7)(6,7)},TE={(1,6), (6,5) }

第三轮

S={1,6,5},U={2,3,4,7},WE={(1,2),(1,7)(6,7),(5,7)(5,4)}选择权值最小的(6,7)

S={1,6,5},U={2,3,4,7},WE={(1,2), (5,4)}TE={(1,6), (6,5),(6,7)}

第四轮

S={1,6,5,7}U={2,3,4}WE={(1,2) (5,4),(7,2),(7,3),(7,4)},选择权值最小的(7,3)

S={1,6,5,7}U={2,3,4} WE={(1,2) (5,4),(7,2), (7,4)}, TE={(1,6), (6,5),(6,7)(7,3)}

第五轮

S={1,6,5,7,3}U={2,4}WE={(1,2) (5,4),(7,2), (7,4),(3,4),(3,2)},选择权值最小的(3,4)

S={1,6,5,7,3}U={2,4}WE={(1,2) (5,4),(7,2), (7,4) , (3,2)}, TE={(1,6), (6,5),(6,7)(7,3),(3,4)}

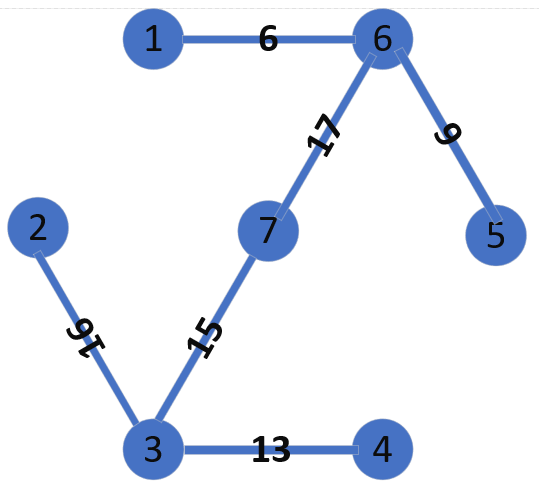
第六轮

S={1,6,5,7,3,4}U={2}WE={(1,2) ,(7,2), (3,2) },选择权值最小的(3,2)

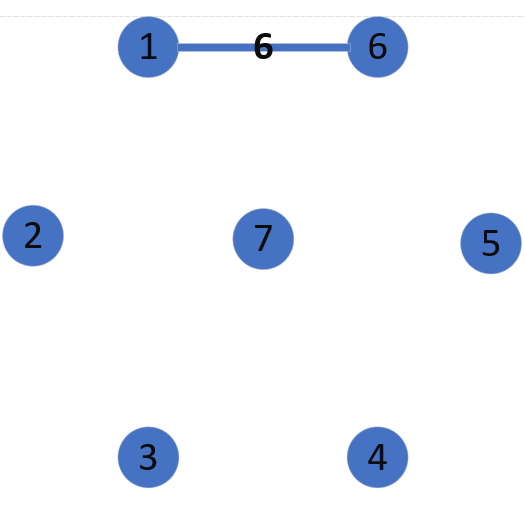
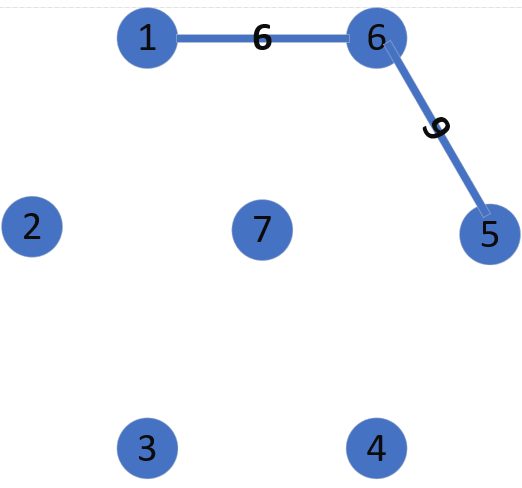
S={1,6,5,7,3,4}U={2}WE={(1,2),(7,2)},TE={(1,6),(6,5),(6,7)(7,3),(3,4),(3,2)}

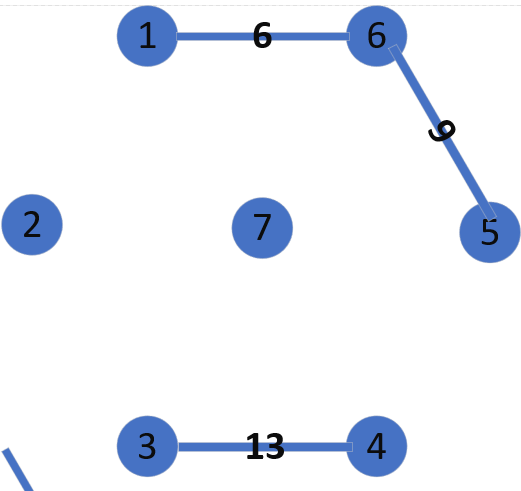
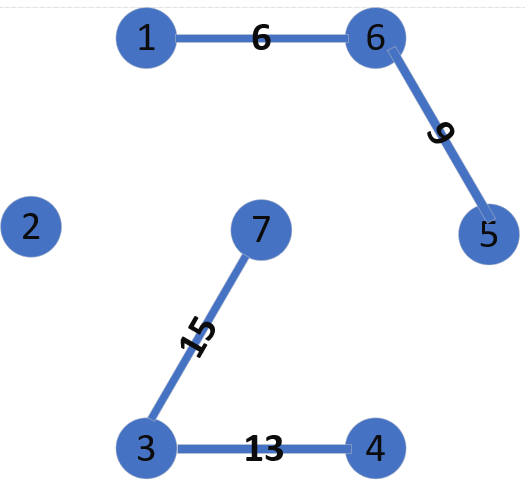
第七轮

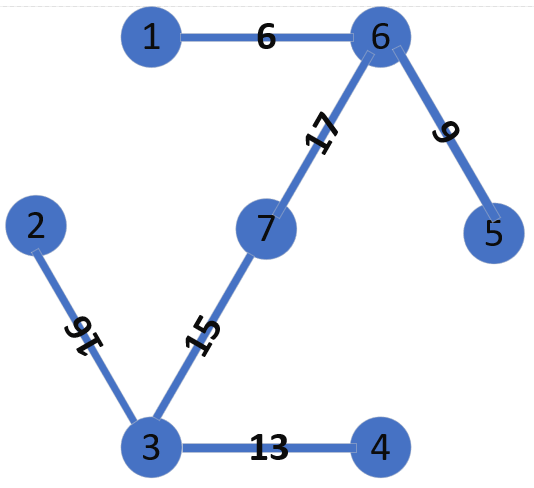
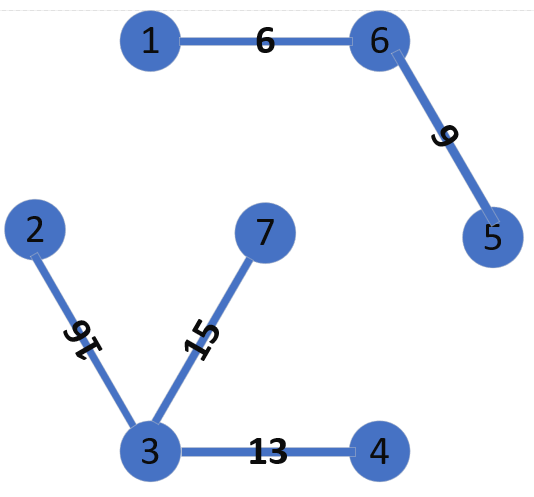
S={1,6,5,7,3,4,2}U={ }WE={ }, TE={(1,6), (6,5),(6,7)(7,3),(3,4),(3,2)};



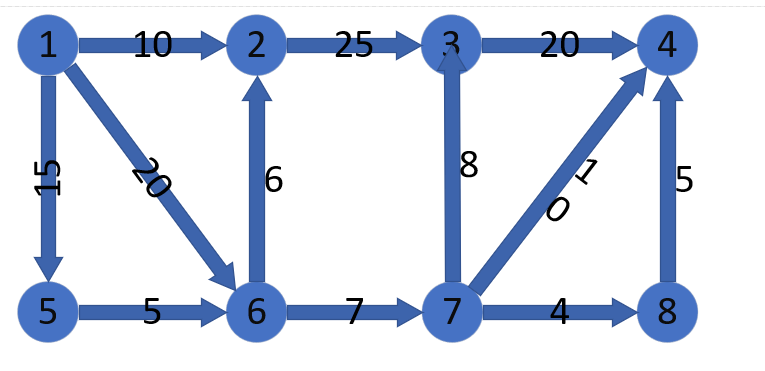
Kruskal算法

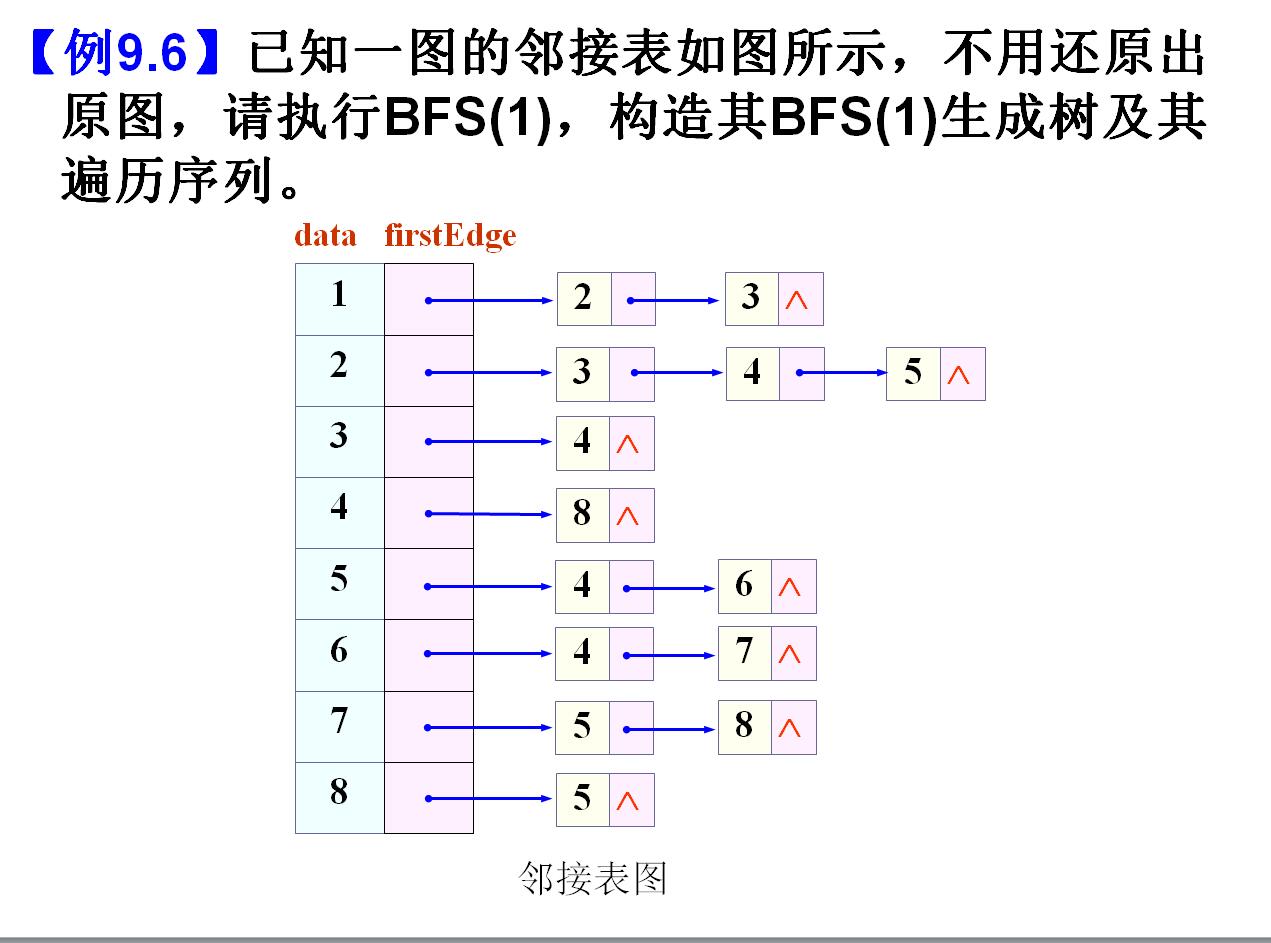


6.28对下面的图求出从顶点1到其余各顶点的最短路径

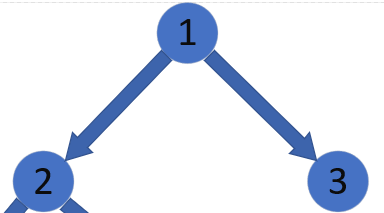


|  |  |  |  |
| --- | --- | --- | --- |
| 操作 | 图 | 各点最短路径 | 路径长度 |
| 标出顶点1可直接到达的路径以及长度 |  | 1 （）  2（1,2）  3 （）  4（）  5（1,5）  6（1,6）  7（）  8（） | 0  10  ∞  ∞  15  20  ∞  ∞ |
| 顶点2距离最近，修改顶点2  以及其后继路径长度 |  | 1 （）  2（1,2）  3 （1,2,3）  4（）  5（1,5）  6（1,6）  7（）  8（） | 0  10  35  ∞  15  20  ∞  ∞ |
| 顶点6距离最近，修改顶点6  以及其后继路径长度 |  | 1 （）  2（1,2）  3 （1,2,3）  4（）  5（1,5）  6（1,6）  7（1,6,7）  8（） | 0  10  35  ∞  15  20  27  ∞ |
| 顶点7距离最近，修改顶点7  以及其后继路径长度 |  | 1 （）  2（1,2）  3 （1,2,3）  4（1,6,7,4）  5（1,5）  6（1,6）  7（1,6,7）  8（1,6,7,8） | 0  10  35  37  15  20  27  31 |
| 顶点8距离最近，修改顶点8  以及其后继路径长度 |  | 1 （）  2（1,2）  3 （1,2,3）  4（1,6,7,8,4）  5（1,5）  6（1,6）  7（1,6,7）  8（1,6,7,8） | 0  10  35  36  15  20  27  31 |
|  |  |  |  |

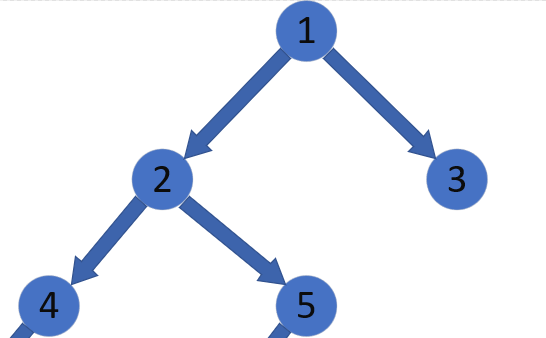
## 9.6



遍历序列为，1 2 3 4 5 8 6 7

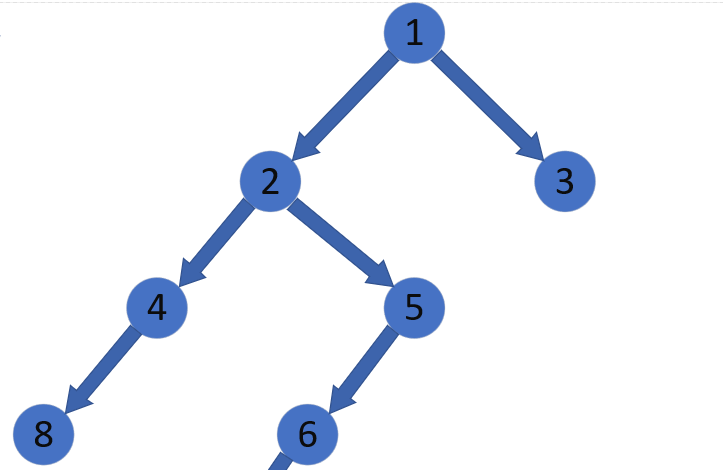


访问顶点1及其邻接点2,3



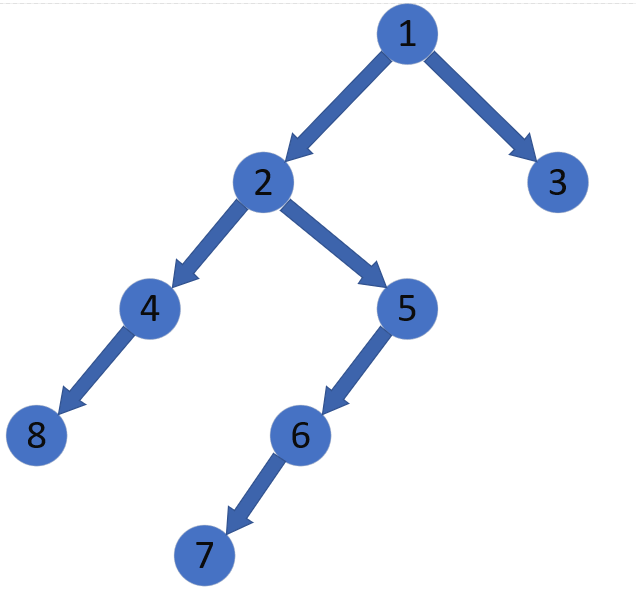
访问顶点2，及其邻接点3，4 ,5，顶点3 已被访问

访问顶点3，因为3的邻接点4已经遍历所以不再访问



访问顶点4及其邻接点8

访问顶点5及其邻接点4,6, 顶点4已经被访问



访问顶点8及其邻接点5 ，顶点5已经被访问

访问顶点6及其邻接点4，7，顶点4已经访问过了

访问顶点7，及其邻接点5 ,8,顶点5，8已经访问过；

最终生成树

