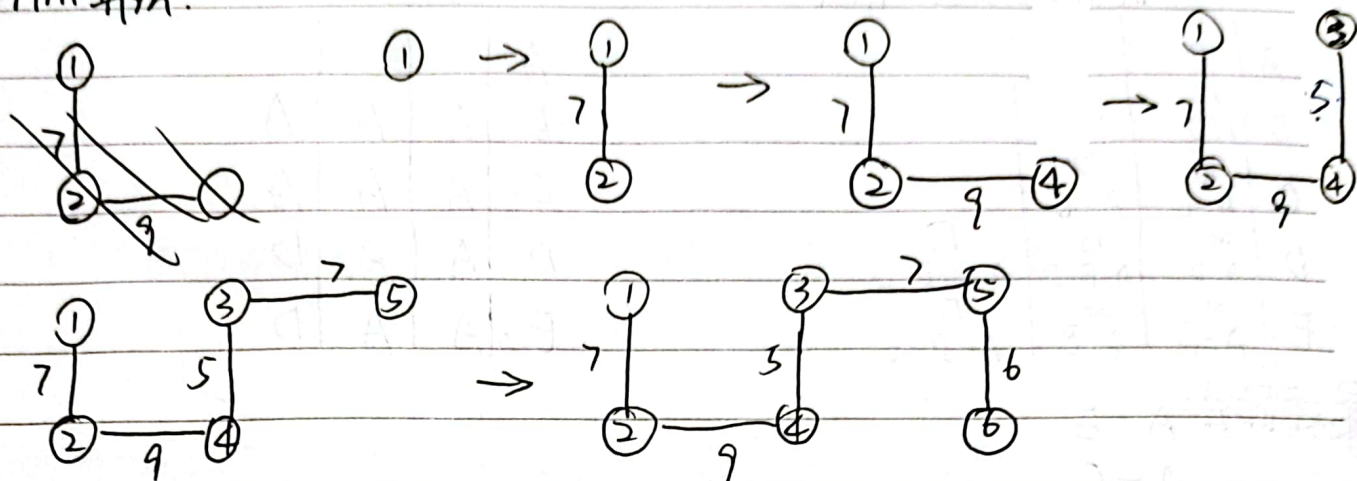
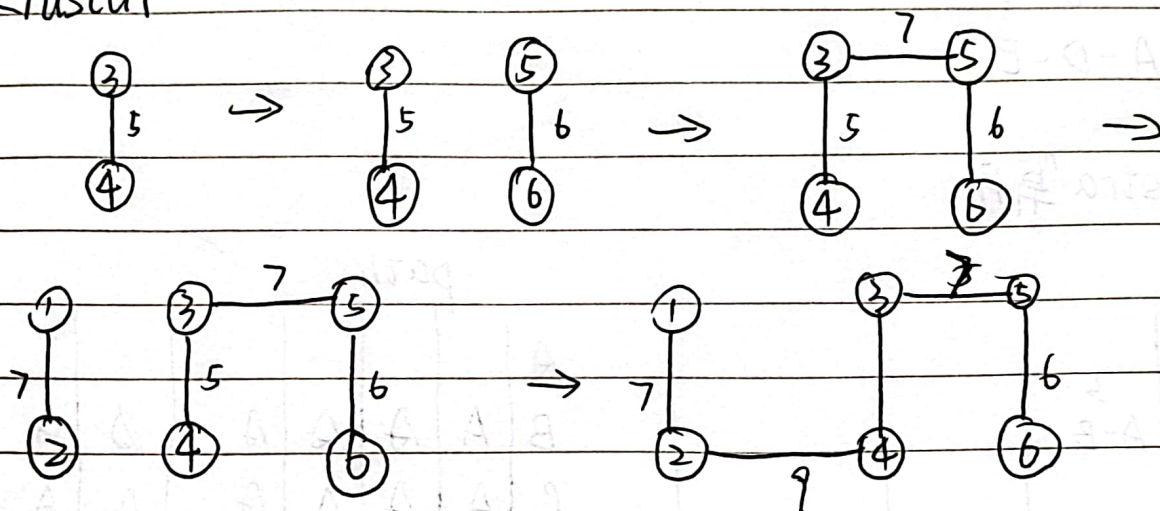


1.11) Prim算法.



12) Kruscal



2. ① V_A 出发

V	V_A	V_B	10.
V_A	V_C	18	
V_A	V_D	∞	
V_A	V_E	∞	

② 考虑经过 V_B .

V	V_A	V_B	10.
V_A	V_C	18.	
V.	V_A	V_B	V_D 15
V_A	V_E	∞	

③ 考虑经过 V_D .

V	V_A	V_B	10
V	V_A	V_D	15
V	V_A	V_D	V_C 17.
V.	V_A	V_D	V_E 17.

距离值数组: dist.

A	0		
B	10		
C	18	18	
D	∞	15	17
E	∞	∞	17
	A-B	A-C	A-B-D-C
	A-D	A-B-D	A-B-D-C
	A-E	A-E	A-D-E

路径path.

A			
B	A	A	A
C	A	A	A
D	A	B	D
E	A	A	D

最短路径:

A-B

A-C

A-B-D-C

A-D-E

3. a. Dijkstra 算法

dist.

A	0			
✓ B	5	5		
✓ C	3			
D	∞	10	10	10
✓ E	∞	10	8	7
F	∞	∞	∞	∞
✓ G	∞	∞	6	
	A-B	A-C	A-B-E	A-B-G-E
	A-D	A-C-D	A-B-E	A-B-G-E
	A-E	A-C-E	A-B-E	A-B-G-E
	A-F	A-F	A-F	A-F
	A-G	A-G	A-B-G	

path

A						
B	A	A	A	A	A	A
C	A	A	A	A	A	A
D	A	C	C	C	E	E
E	A	C	B	G	G	G
F	A	A	A	A	E	E
G	A	A	B	B	B	B

接上表.

	9	9
D	A-B-G-E-D	A-B-G-E-D
✓ F	8	
	A-B-G-E-F	

最短路径: A-B 5

A-B-G 6.

A-C 3

A-B-G-E-D 9

A-B-G-E 7.

A-B-G-E-F 8

b. 1° B指向的邻居 $B \rightarrow C$ $B \rightarrow G$ $B \rightarrow E$.

2° C.G.E指向的邻居 $B \rightarrow C \rightarrow D$

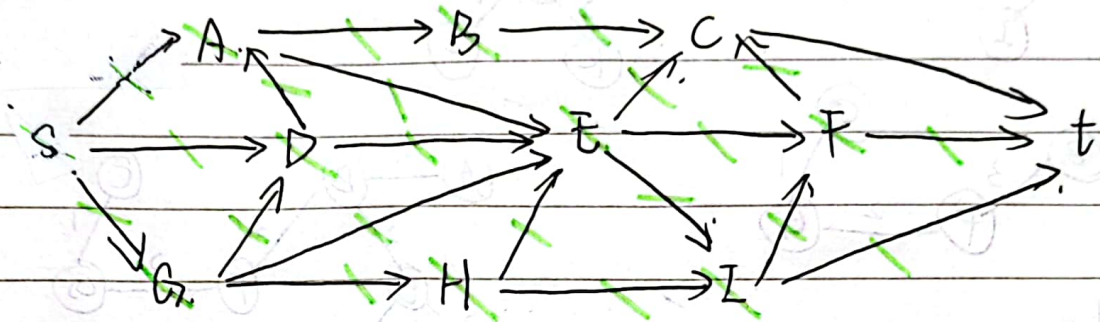
$B \rightarrow E \rightarrow D$

$B \rightarrow E \rightarrow F$

3° D.F指向的邻居 $B \rightarrow C \rightarrow D \rightarrow A$ $B \rightarrow E \rightarrow D \rightarrow A$.

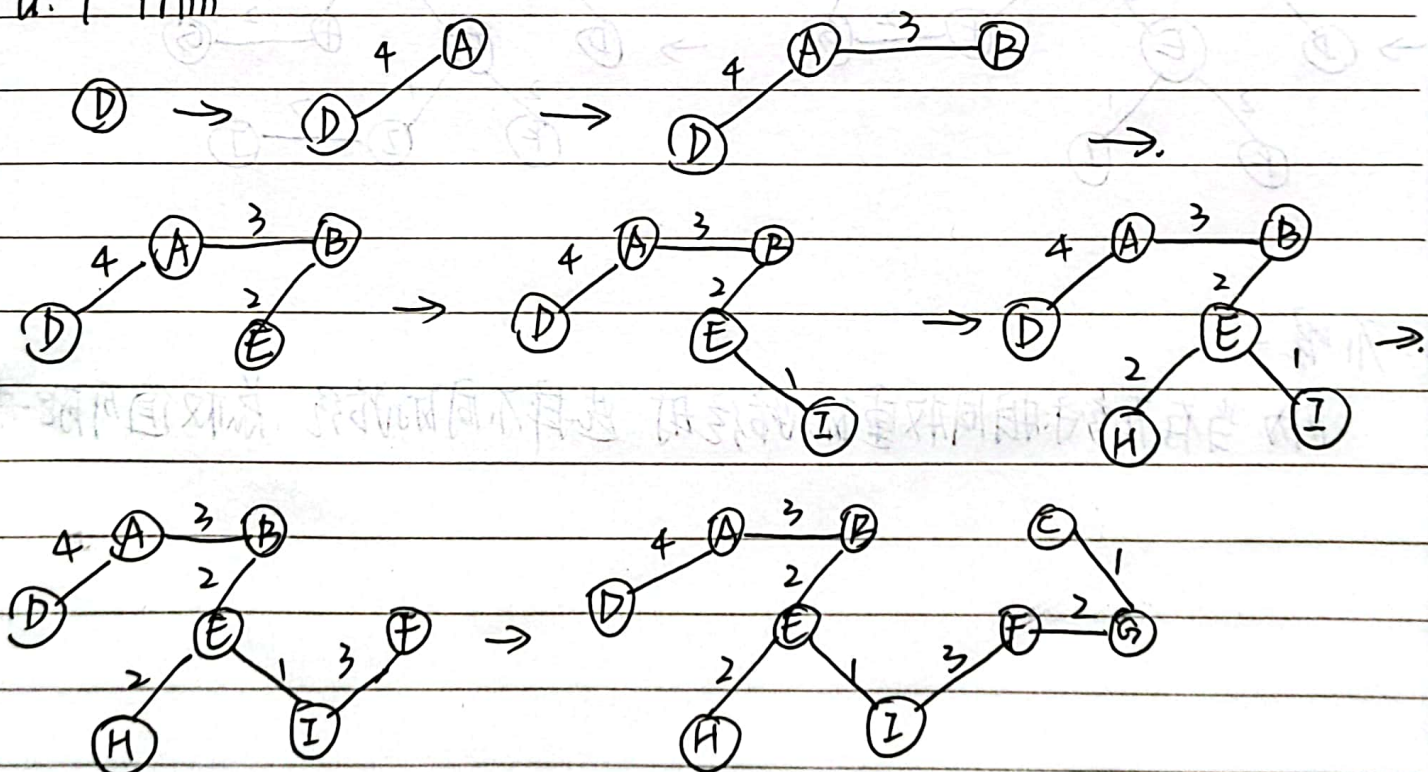
~~$B \rightarrow E \rightarrow F$~~

4. 依次找到入度为0的顶点. 删除该顶点及其所有出边

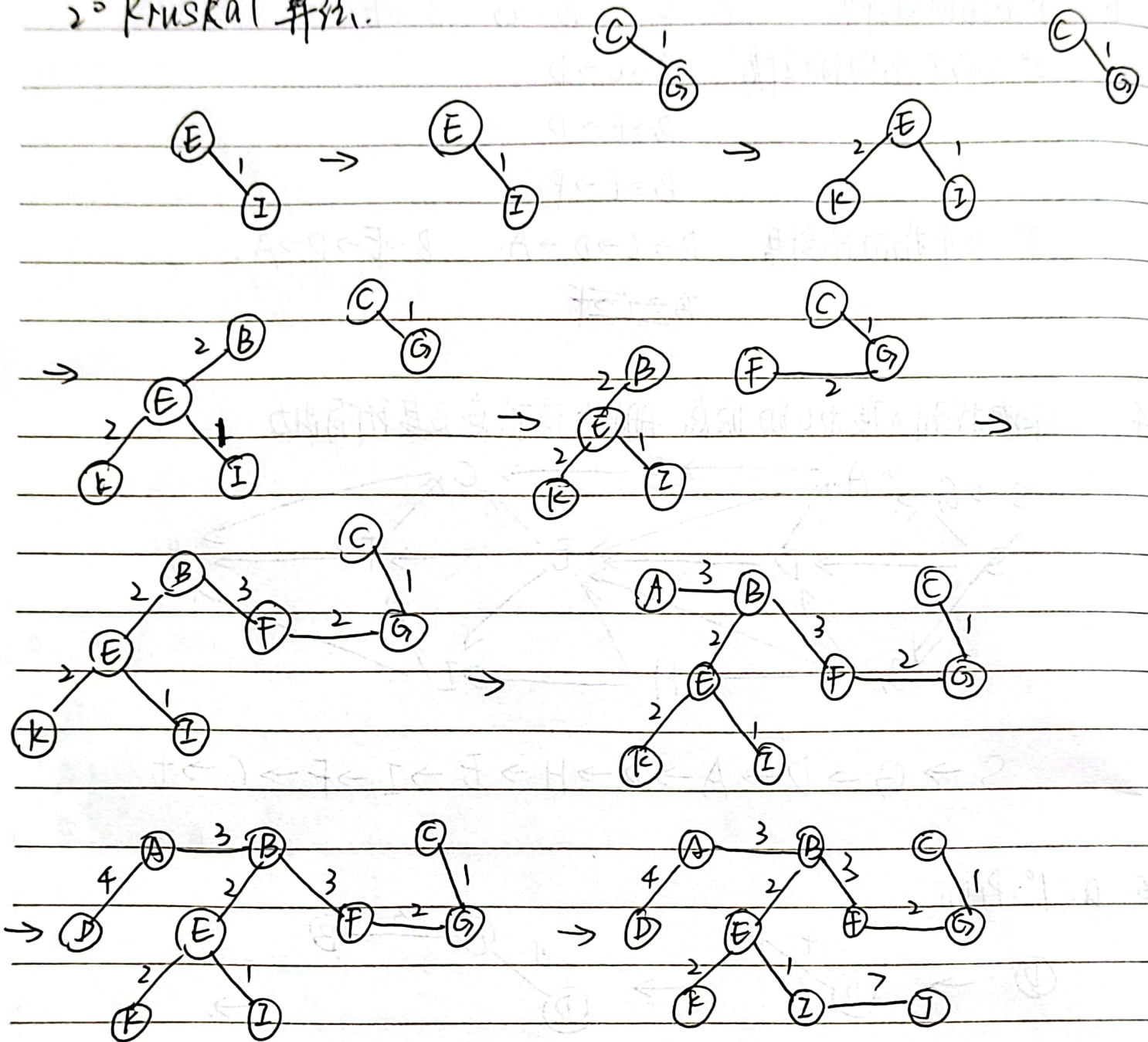


$S \rightarrow G \rightarrow D \rightarrow A \rightarrow B \rightarrow H \rightarrow E \rightarrow I \rightarrow F \rightarrow C \rightarrow t$

5. a. 1° Prim



2. Kruskal 算法.



b. 不唯一 -

因为当存在多条相同权重的路径时, 选择不同的路径, 总权值可能一样.