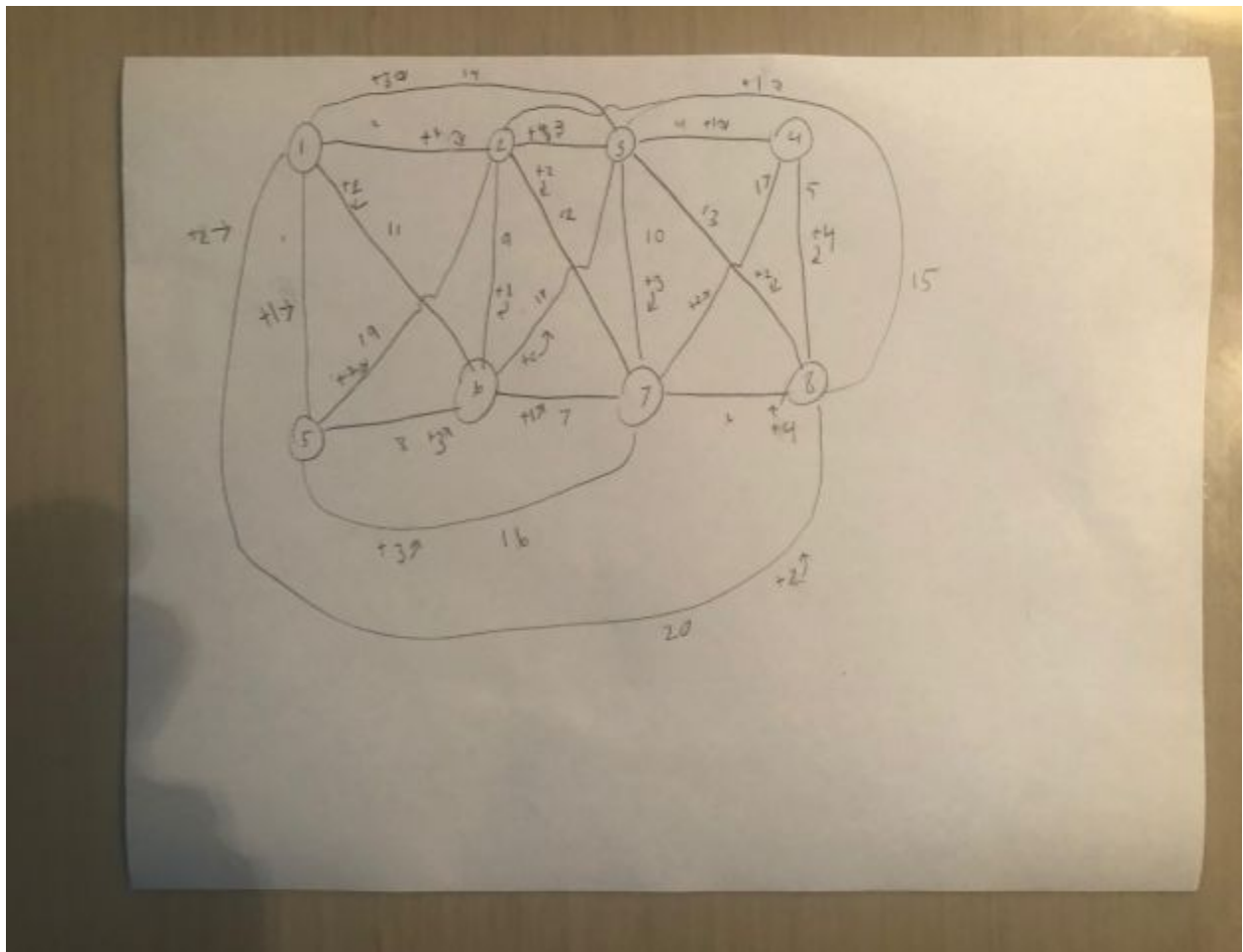


Graph



Table

Graph Table

Node 1

Edges	Weight
2	1
3	3
5	1
6	2
8	2

5

E	W
1	1
2	2
4	3
7	3

2

E	W
1	1
3	4
5	2
6	1
7	2
8	1

6

E	W
1	2
2	1
3	2
5	3
7	1

3

E	W
1	3
2	4
4	1
6	2
7	3
8	2

7

E	W
2	2
3	3
4	2
5	3
6	1
8	4

4

E	W
3	1
7	2
8	4

8

E	W
1	2
2	1
3	2
4	4
7	4

Prim's Algorithm

Table

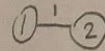
Prim's Algorithm

Graph

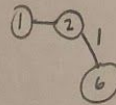
1.



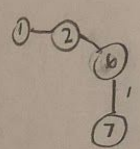
2.



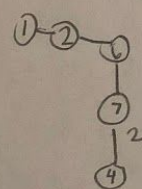
3.



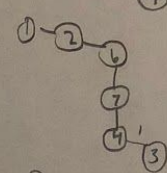
4.



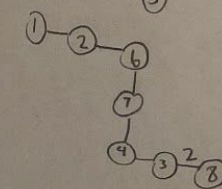
5.



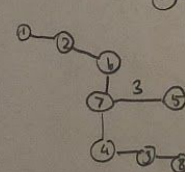
6.



7.

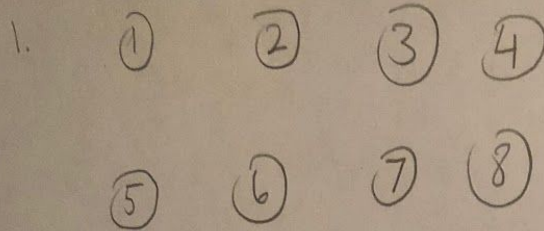


8.



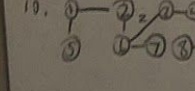
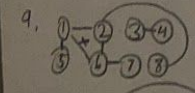
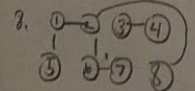
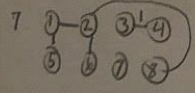
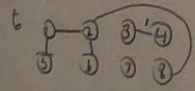
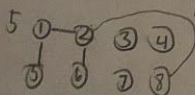
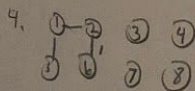
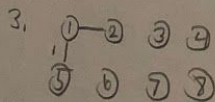
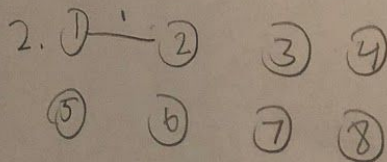
Kruskal's Algorithm

Kruskal's Algorithm



Edges

#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	1,2	1,5	2,3	2,4	3,4	3,5	4,5	4,6	5,6	5,7	6,7	6,8	7,8	7,9	8,9	8,10	9,10	10,11	11,12	12,13
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1



cycle We skip 1, 6, 1, 8, 2, 5, 2, 7, because they form cycles

Dijkstra's Algorithm

1 → 2 1.

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

Queue

- 0 + 1 = 1
- 0 + 3 = 3
- 0 + 4 = 4
- 0 + 2 = 2
- 0 + 2 = 2

Visited: 1 Current: 1

	1	2	3	4	5	6	7	8
2.	0	1	3	∞	1	2	2	
3								

Queue

- 5, 1
- 6, 2
- 8, 2
- 3, 3
- 7, 3

Visited: 1, 2 Current: 2

	1	2	3	4	5	6	7	8
3	0	1, 2	3, 3	∞	1, 1, 5	2, 1, 2	2	
4								

Queue

- 4, 2
- 8, 2
- 3, 3
- 7, 3

Visited: 1, 2 Current: 1, 5

	1	2	3	4	5	6	7	8
4	0	1, 2	3, 3	∞	1, 1, 5	2, 1, 2	3, 1, 2, 7	2, 1, 2
5								

Visited: 1, 2 Current: 1, 5

5. 1 2 3 4 5 6 7 8
 0 1-2 3-3 4-3-4 1-5 2-6 3-2-7 2-8

Queue	Visited	Current	W
8,3	1,2	1,8	2
7,3	1,5		
4,6	1,6		
	1,8		

6. 1 2 3 4 5 6 7 8
 0 1-2 3-3 4-3-4 1-5 2-6 3-2-7 2-8

Queue	Visited	Current	W
7,3	1,2	1-3	3
4,4	1,5		
	1,6		
	1,8		
	1,3		

7. 1 2 3 4 5 6 7 8
 0 1-2 3-3 4-3-4 1-5 2-6 3-2-7 2-8

Queue	Visited	Current	W
4,4	1,2	1-2-7	3
	1,5		
	1,6		
	1,8		
	1,3		
	1,2,7		

8.	1	2	3	4	5	6	7	8
	0	1-2	3-3	4-3-4	1-5	2-6	3-2-7	2-8
	Queue		visited	current	W			
	Done		1, 4	1-3-4	4			
			1, 5					
			1, 6					
			1, 8					
			1, 3					
			1, 2, 7					
			1, 3, 4					

The shortest route from 1 to the other nodes are the following

Destination	1	2	3	4	5	6	7	8
Cost	0	1	3	4	1	2	3	2
Path	1-1	1-2	1-3	1-3-4	1-5	1-6	1-2-7	1-8