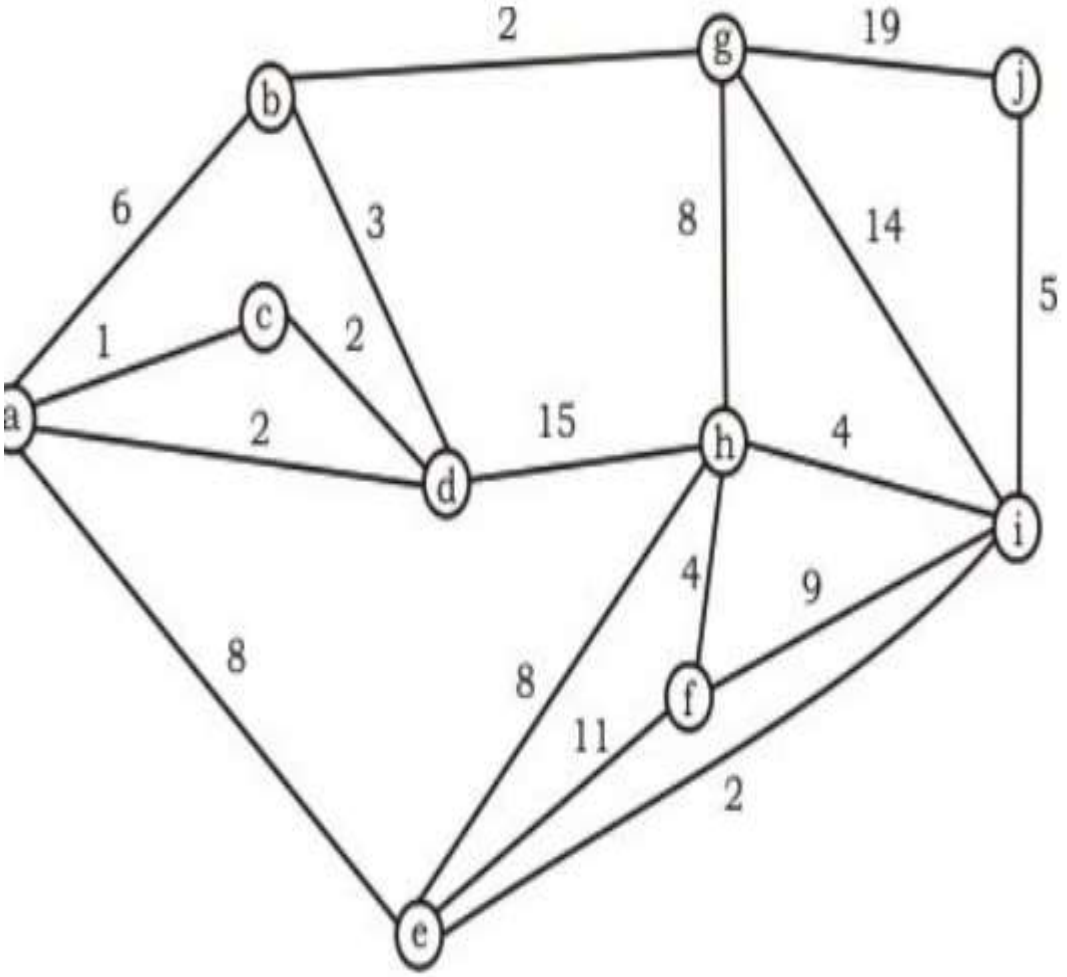
CS501
Mansi Shah
(19526)
(hw7:wk7)

Q4: Please use MST approached propsed by Prim & Kruskal to find the MST of the following diagram and then compare their Time Complexity.



(kruskal Algoritham)

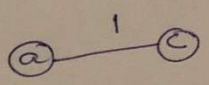
The graph contains 10 vertices and 18 edges so the minimum spanning Tree formed will be having (10-1) = 9 edges

After sorting

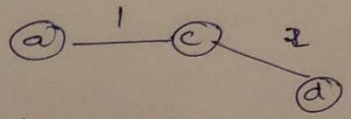
Weight	Src	Destination
1	a	c
2	c	d
2	a	d
2	e	i
2	b	g
3	b	d
4	h	i
4	h	f
5	i	j
6	a	b
8	a	e
8	g	h
8	h	e
9	f	i
11	e	f
14	g	i
15	d	h
19	g	j

Now pick all edges one by one from sorted list of edges

Step 1 : Pick edges a to c
No cycle formed included it

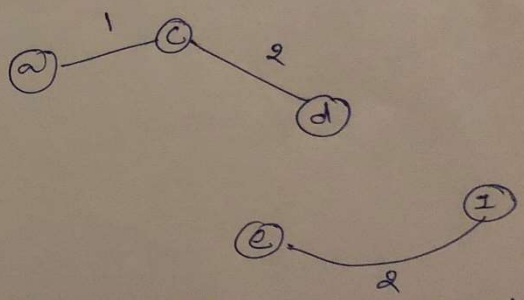


Step 2: Pick c to d
no cycle is formed, included it

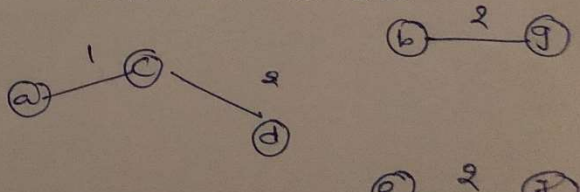


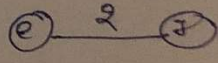
Step 3: pick a to d cycle formed skip it.

Step 4: Pick e to i, no cycle is formed included it.

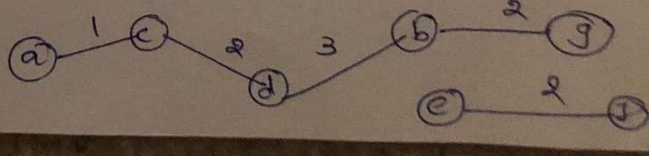


Step 5: Pick b to g, no cycle is formed, included it

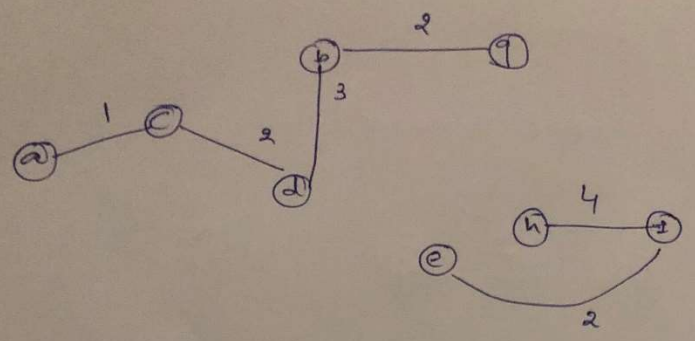




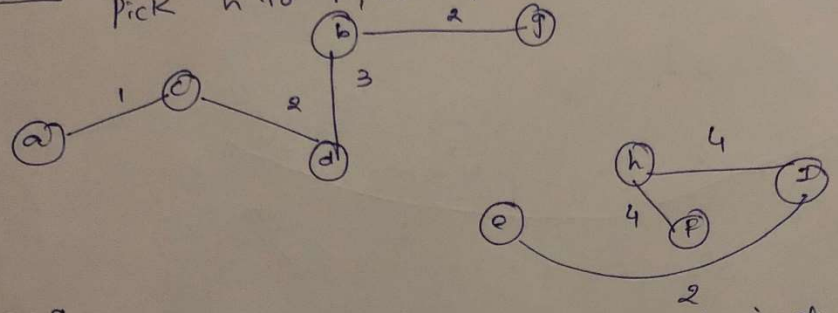
Step 6: Pick b to d, no cycle is formed, included it



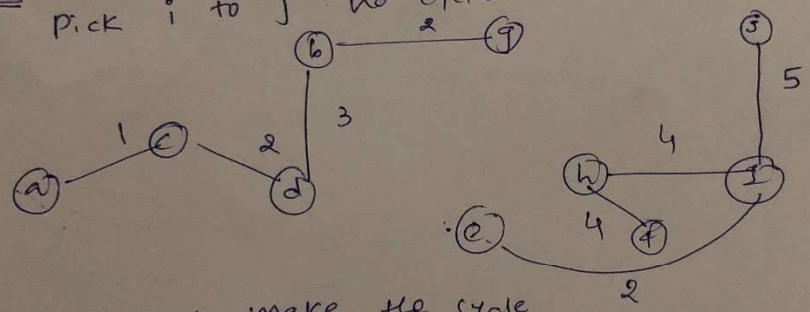
Step : 7 Pick h to I, no cycle formed, included it



Step : 8 Pick h to f, no cycle formed, included it

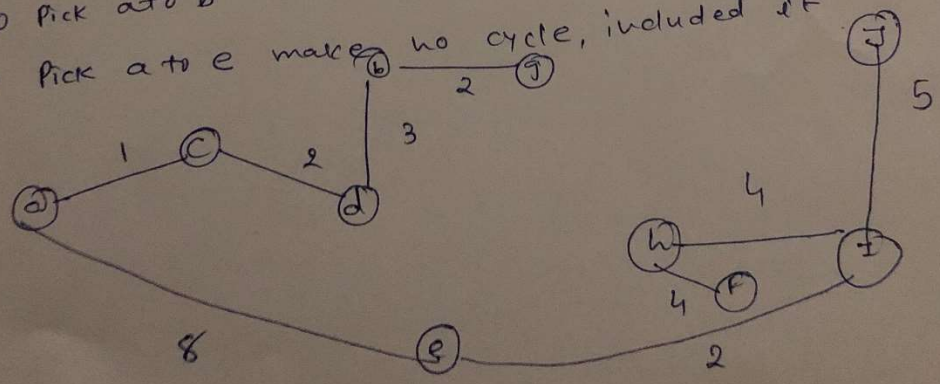


Step : 9 Pick i to j no cycle formed, included it.



Step: 10 Pick a to b make the cycle

Step: 11 Pick a to e make no cycle, included it

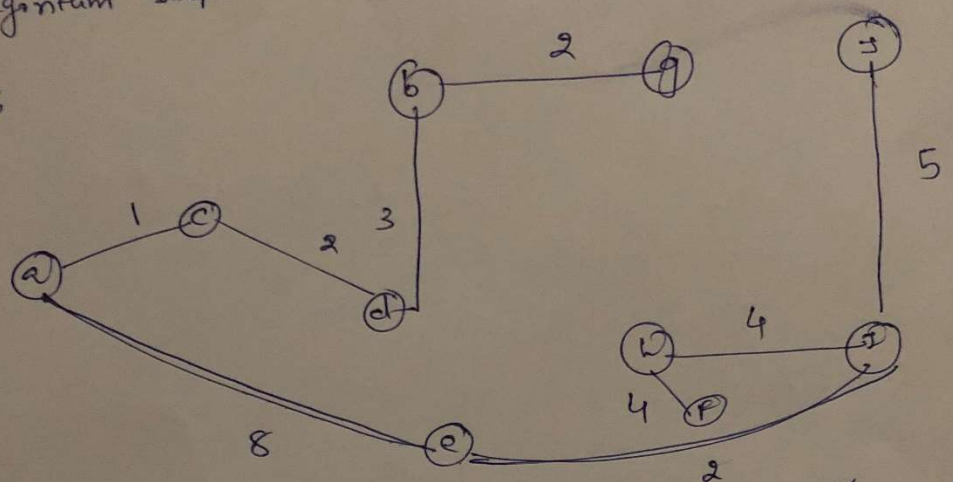


Step: 12 pick g to h make cycle skip it

Step: 13 pick h to e make cycle skip it.

Since the number of edges included equals (V-1), the algorithm stops here.

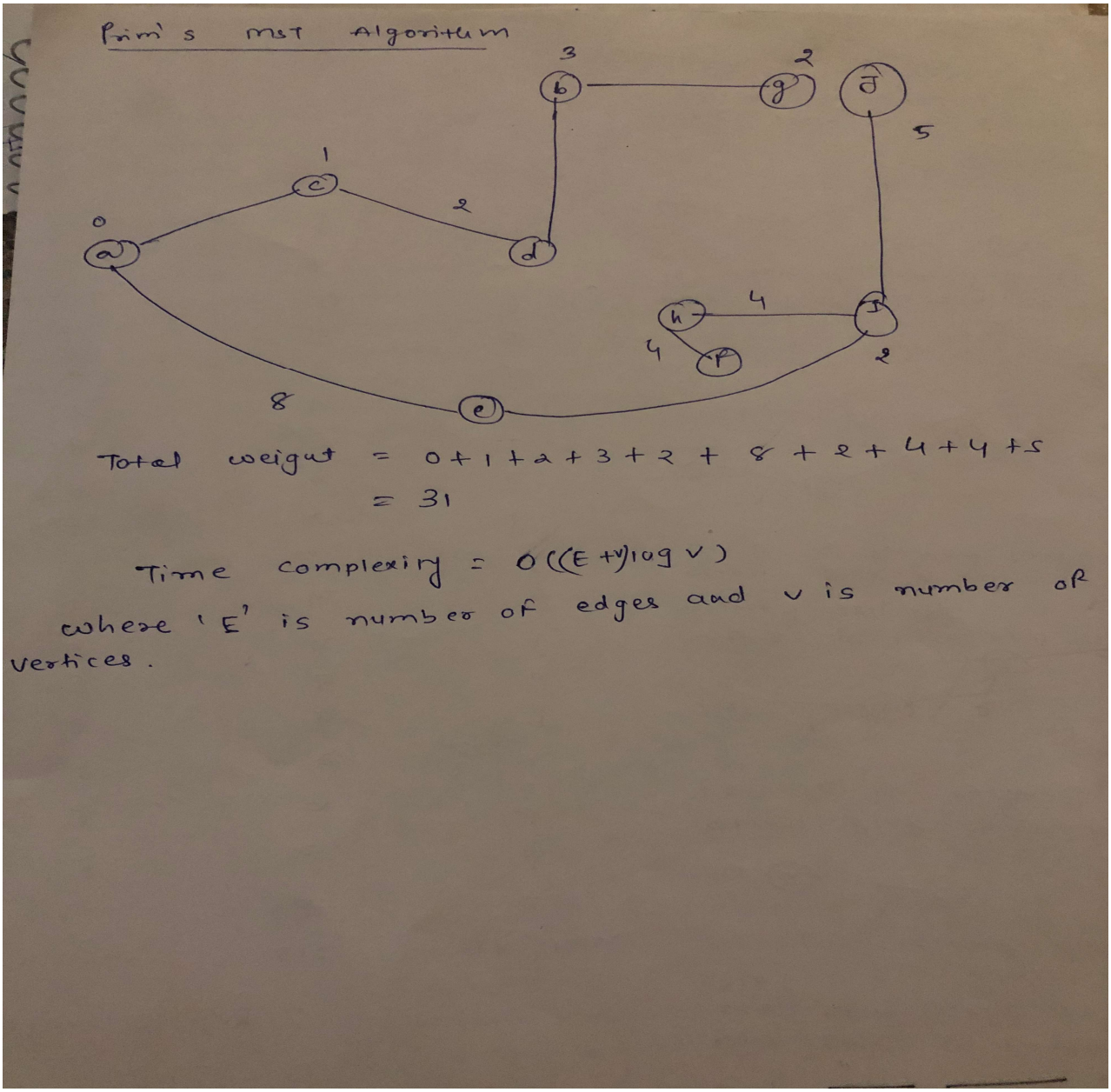
so;



Total weight = 1+2+2+2+3+4+4+5+8

= 31

Time complexity : $O(E^* \log V)$.



Use Prim's algorithm when you have a graph with lots of edges.

For a graph with V vertices E edges, Kruskal's algorithm runs in O(E log V) time and Prim's algorithm can run in O(E + V log V) amortized time, if you use a Fibonacci Heap 20.

Prim's algorithm is significantly faster in the limit when you've got a really dense graph with many more edges than vertices. Kruskal performs better in typical situations (sparse graphs) because it uses simpler data structures.