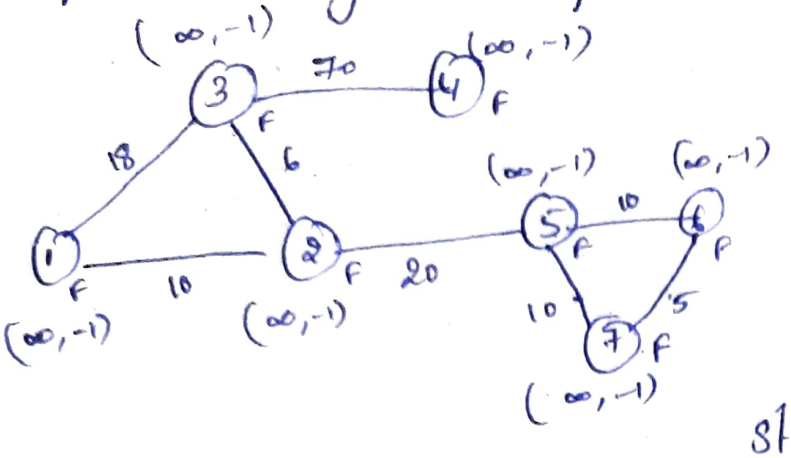
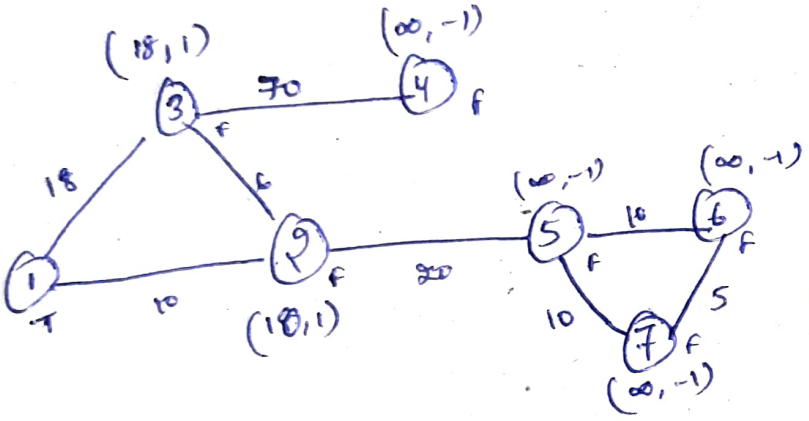
Prim's Algo Example:

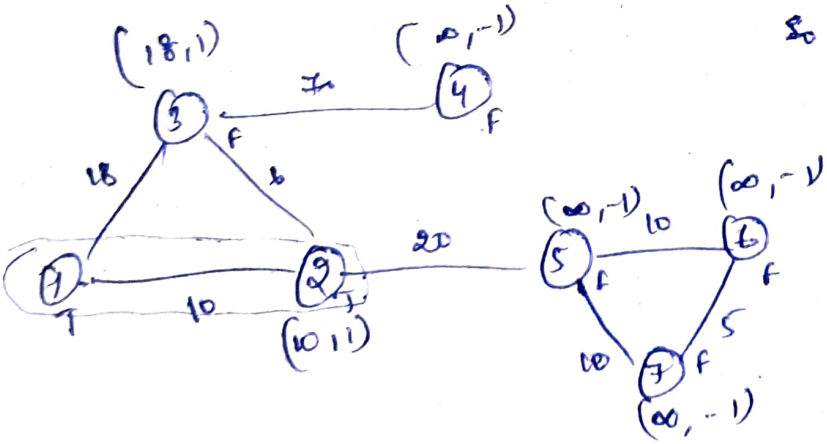


Step ①:
Initialize. i = 1 ---n
visited [i] = false
distance_Tv[i] = ∞
Neighbour_Tv[i] = -1

Step ② visited[i] = True.
update all neighbours of it
j are the edges of 1
Neighbour[j] = 1
distance_Tv[j] = weight of (i,j)



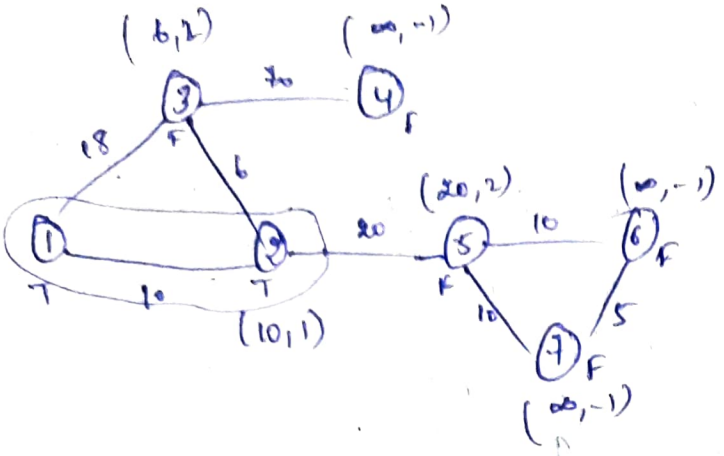
Step ③
chose u such that
visited[u] = false and
distance_Tv[u] is min
So picking ②
visited[2] = True



Step ④
for all neighbours of ②
which are not visited

2's neighbours ③ and ⑤ which are not visited
③ → (18,1) → (6,2) (18 > 6 so)
⑤ → (∞,-1) → (20,2) (∞ > 20 so)

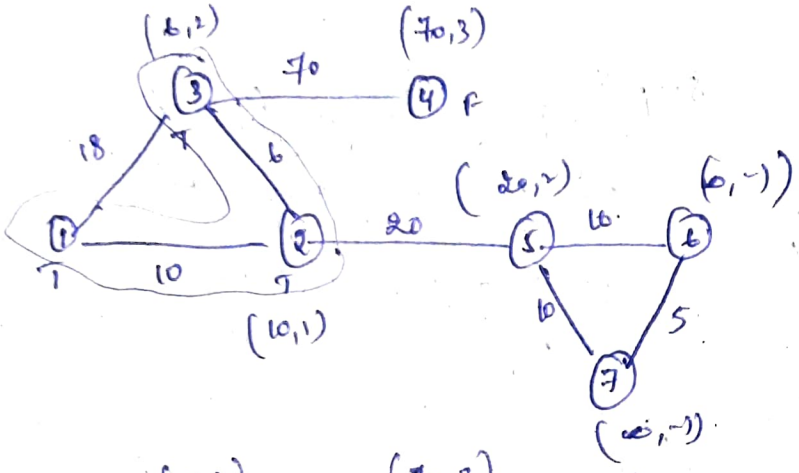
If distance_Tv[v] > weight(u,v):
Distance_Tv[v] = weight(u,v)
neighbour_Tv[v] = 4



Repeat Step ③ and ④

u ⇒ 3 visited[3] = True

④ → (∞,-1) → (70,3)



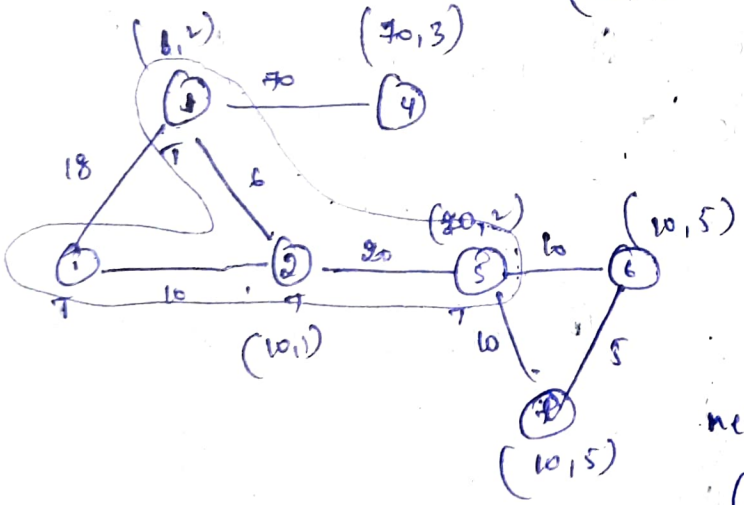
Repeat step ① and ④

u = 5 visited[5] = True

neighbours

6 → (∞,-1) → (10,5)

7 → (∞,-1) → (10,5)



Repeat step ③ and ④

u = 6 or 7

So picking ⑦

neighbours

⑥ → (10,5) → (5,7)

(10 > 5 so)

This procedure Continues and final result is