Write pseudo code for Kruskal's algorithm.

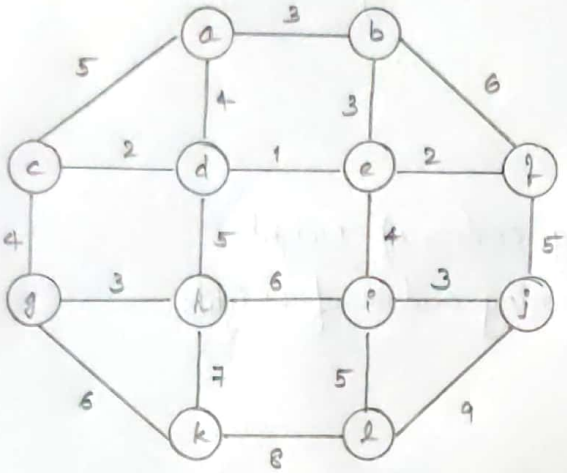
Kruskal's Algorithm

```
procedure kruskal (G, w)
Input: A connected undirected graph G=(V,E) with edge weight we
Output: A minimum spanning tree defined by the edges X.
for all u∈V:
   makeset(u)
X={}
Sort the edges E in non-decreasing order of weight
for all edges {u,v} ∈E, in increasing order of weight:
   if find(u) ≠ find(v):
      add edge {u,v} to X
      union (u,v)
procedure makeset (x)
π(x)=x
rank(x)=0
function find(x)
while x≠π(x): x=π(x)
return x
procedure union (x,y)
rx = find(x)
ry = find(y)
if rx = ry : return
if rank(rx) > rank(ry):
   π(ry) = rx
else:
   π(rx) = ry
```

if rank(x) = rank(y) : rank(y) = rank(y) + 1

Use the algorithm to find the minimum cost spanning tree for the graphs given below.

①



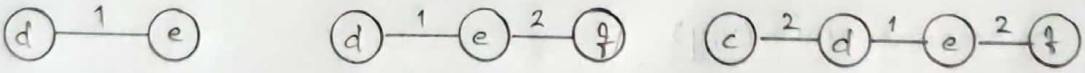
ans Step 1:-

Edge	{d,e}	{e,f}	{c,d}	{a,b}	{b,e}	{g,h}	{i,j}	{a,d}	{c,g}	{e,i}	{a,c}
Weight	1	2	2	3	3	3	3	4	4	4	5

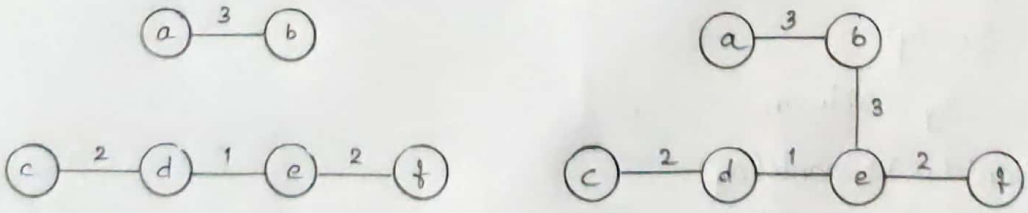
{d,h}	{f,j}	{i,l}	{b,f}	{h,i}	{g,k}	{h,k}	{k,l}	{j,l}
5	5	5	6	6	6	7	8	9

Step 2:- Smallest edge = {d,e} → Edge with smallest weight

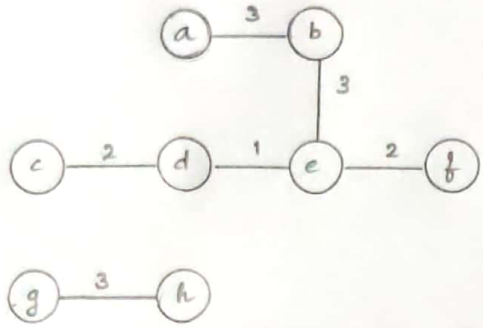
Step 3:- (i) (ii) (iii)



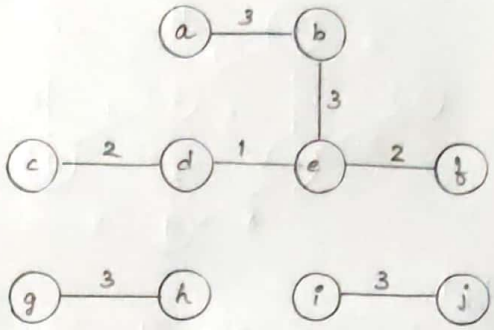
(iv) (v)



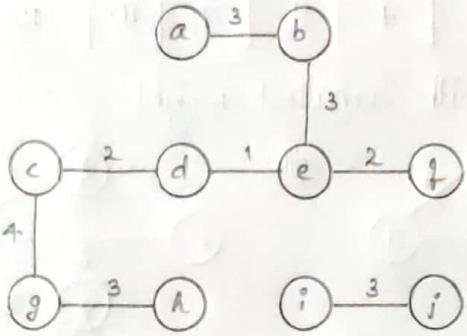
(vi)



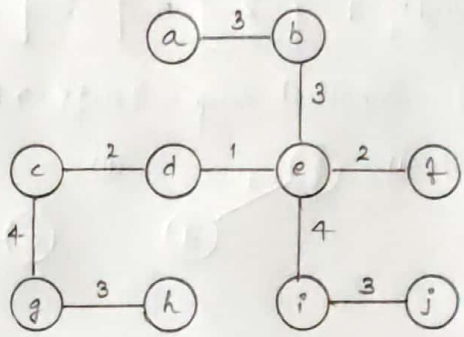
(vii)



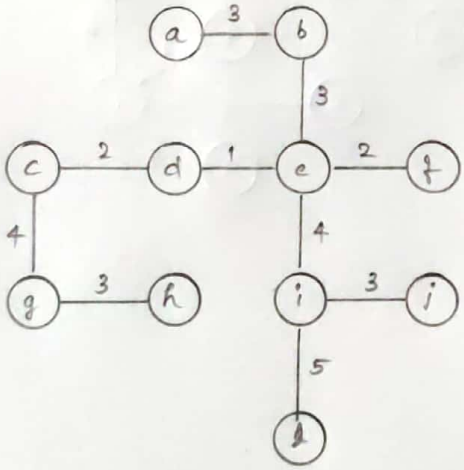
(viii)



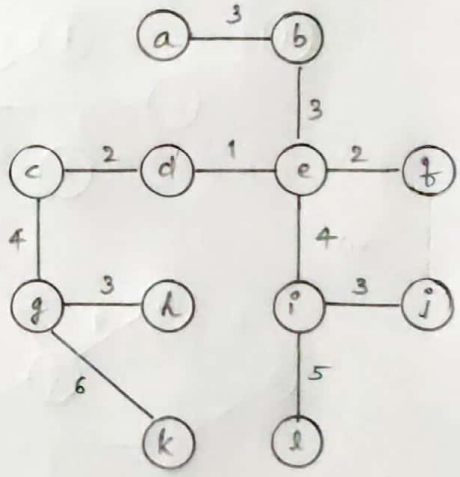
(ix)



(x)



(xi)

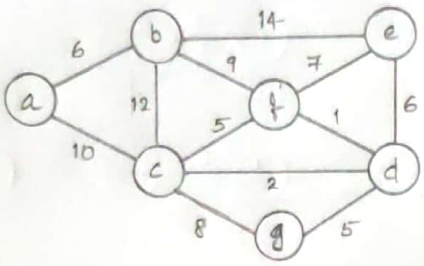


Step 4:- No. of vertices - 1 = 12 - 1 = 11 edges have been obtained. So we stop.

Minimum weight = 3+3+2+1+2+4+4+3+3+6+5

= 36

②

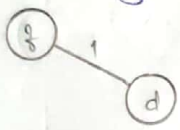


ans. Step 1:-

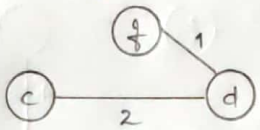
Edge	{d, f}	{c, d}	{c, f}	{d, g}	{a, b}	{d, e}	{e, f}	{c, g}	{b, f}	{a, c}	{b, c}	{b, e}
Weight	1	2	5	5	6	6	7	8	9	10	12	14

Step 2:- Smallest edge = {d, f} → Edge with smallest weight

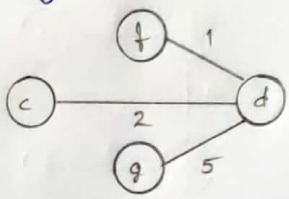
Step 3:- (i)



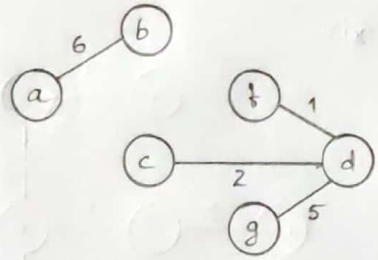
(ii)



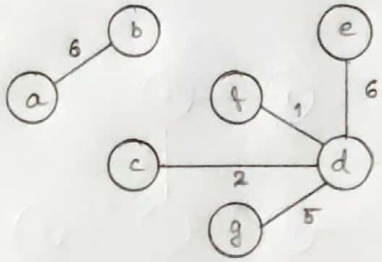
(iii)



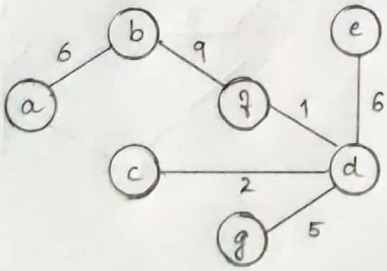
(iv)



(v)



(vi)



Step 4:- No. of vertices - 1 = 7 - 1 = 6 edges have been obtained. So we stop.

Minimum weight = 1+2+5+6+6+9

= 29