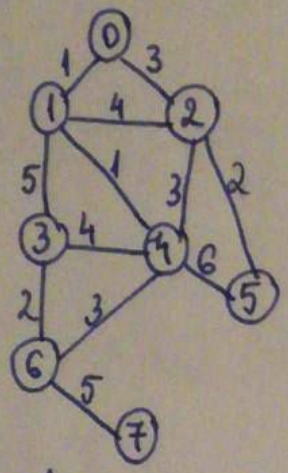
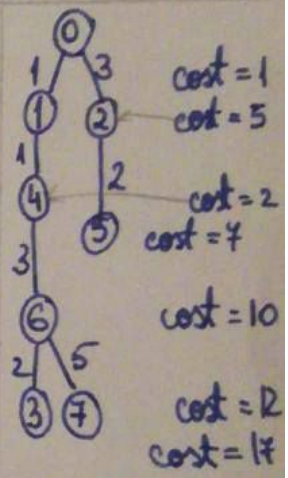
Minimum spanning tree - Prim's Algorithm



Start = 0

Edges sorted:

(0,1) - 1
(1,4) - 1
(3,6) - 2
(2,5) - 2
(0,2) - 3
(2,4) - 3
(4,6) - 3
(1,2) - 4
(3,4) - 4
(1,3) - 5
(6,7) - 5
(4,5) - 6

	Selection process description	selected edge	Vertices	Edges	minimum spanning tree
initialization			{0}	{ }	
iteration 1	We have to choose the edge with the lowest cost between: (0,1) - 1 ; (0,2) - 3.	(0,1)	{0,1}	{(0,1)}	
iteration 2	choose between: (0,2) - 3 ; (1,2) - 4 ; (1,4) - 1 ; (1,3) - 5	(1,4)	{0,1,4}	{(0,1) ; (1,4)}	
iteration 3	choose between: (0,2) - 3 ; (1,2) - 4 ; (1,3) - 5 ; (4,2) - 3 ; (4,5) - 6 ; (4,3) - 4 ; (4,6) - 3	(0,2)	{0,1,4,2}	{(0,1) ; (1,4) ; (0,2)}	
iteration 4	choose between: (1,2) - 4 ; (1,3) - 5 ; (4,2) - 3 ; (4,5) - 6 ; (4,3) - 4 ; (4,6) - 3 ; (2,5) - 2	(2,5)	{0,1,4,2,5}	{(0,1) ; (1,4) ; (0,2) ; (2,5)}	
iteration 5	choose between: (1,2) - 4 ; (1,3) - 5 ; (4,2) - 3 ; (4,5) - 6 ; (4,3) - 4 ; (4,6) - 3	(4,6)	{0,1,4,2,5,6}	{(0,1) ; (1,4) ; (0,2) ; (2,5) ; (4,6)}	
iteration 6	choose between: (1,2) - 4 ; (1,3) - 5 ; (4,2) - 3 ; (4,5) - 6 ; (4,3) - 4 ; (6,3) - 2 ; (6,7) - 5.	(6,3)	{0,1,4,2,5,6,3}	{(0,1) ; (1,4) ; (0,2) ; (2,5) ; (4,6) ; (6,3)}	
iteration 7	choose between: (1,2) - 4 ; (1,3) - 5 ; (4,2) - 3 ; (4,5) - 6 ; (4,3) - 4 ; (6,7) - 5	(6,7) (*)	{0,1,2,5,6,3,7}	{(0,1) ; (1,4) ; (0,2) ; (2,5) ; (4,6) ; (6,3) ; (6,7)}	

(*) The edge (6,7) isn't the edge with the lowest cost between the "candidate" edges, but it is the only one we can choose for not creating a cycle.