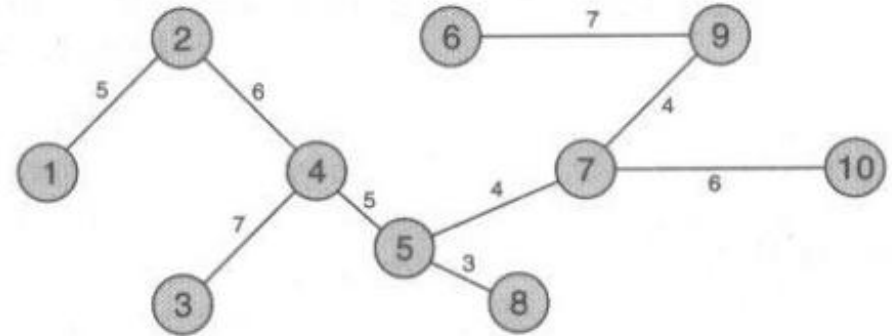
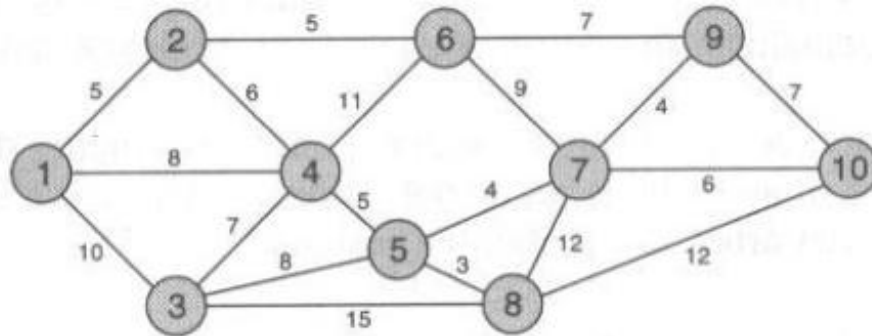


Algoritmo Prim

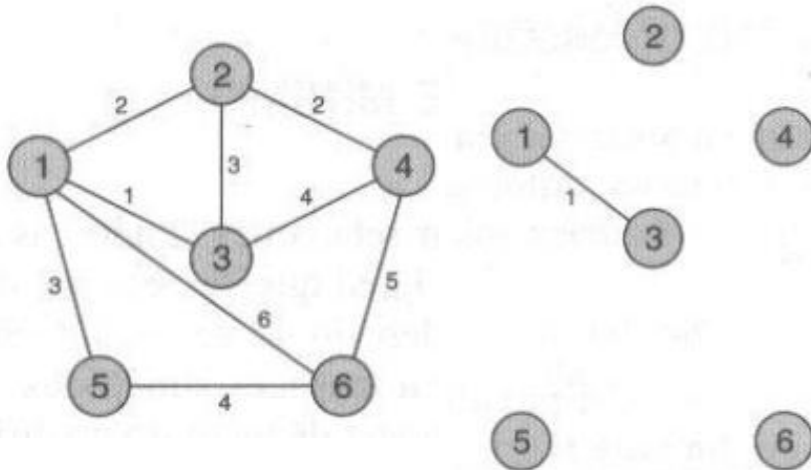


$W = \{1\}$ (1,2)
 $W = \{1, 2\}$ (2,4)
 $W = \{1, 2, 4\}$ (4,5)
 $W = \{1, 2, 4, 5\}$ (5,8)
 $W = \{1, 2, 4, 5, 8\}$ (5,7)
 $W = \{1, 2, 4, 5, 8, 7\}$
 $W = \{1, 2, 4, 5, 8, 7, 9, 10, 3, 6\}$

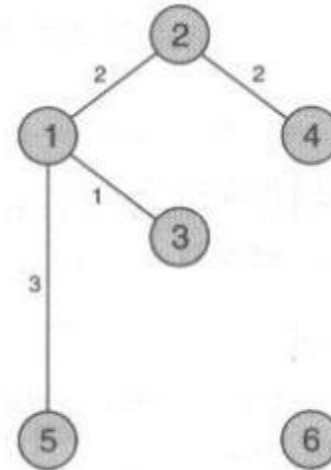
```

arbolExpansion Prim(G, T)
{ G : grafo
  T : conjunto de arcos del árbol de coste mínimo}
var locales
  W: < conjunto de vértices >;
  u,w: vértices;
Inicio
  T <- {}
  V <- {1..n}
  u <- 1
  W <- {u}
  mientras W<> V hacer
    <Encontrar v de V-W tal que (u,v) sea mínimo>
    W <- W+{v}
    T <- T+{(u,v)}
  fin_mientras
fin_arbolExpansion
    
```

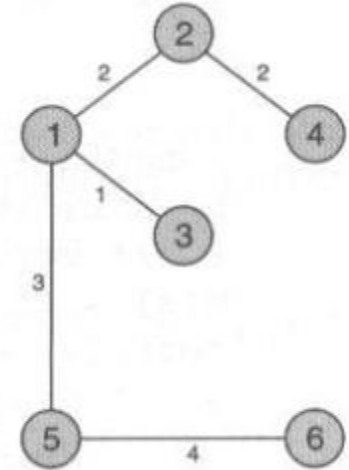
Algoritmo Kruskal



a)



b)



c)

La lista de aristas en orden creciente

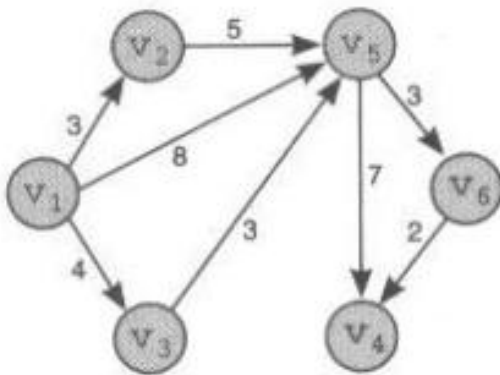
$\{(1,3), (1,2), (2,4), (1,5), (2,3), (3,4), (5,6), (4,5), (1,6)\}$

Si $P(1,5) = 4$, $P(5,6) = 3$, entonces

1	1 3	1 3 2	1 3 2 4 5	1 3 2 4 5 6
2	2			
3				
4	4	4		
5	5	5		
6	6	6	6	

1	1 3 2 4	1 3 2 4	1 3 2 4	1 3 2 4 5 6
2				
3				
4				
5	5 6	5	5 6	
6		6		

Algoritmo Dijkstra



$$C = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 5 \\ 6 \end{matrix} & \begin{vmatrix} \infty & 3 & 4 & \infty & 8 & \infty \\ \infty & \infty & \infty & \infty & 5 & \infty \\ \infty & \infty & \infty & \infty & 3 & \infty \\ \infty & \infty & \infty & 7 & \infty & 3 \\ \infty & \infty & \infty & 2 & \infty & \infty \end{vmatrix} \end{matrix}$$

```

int F[N];
double matPesos[N][N];

typedef struct
{
    int ultimo;
    double distancia;
} EstadoVertice;

EstadoVertice D[N];
  
```

Paso	F	v	D[2]	D[3]	D[4]	D[5]	D[6]
Inicial	1		3	4		8	
1	1, 2	2	3	4		8	
2	1, 2, 3	3	3	4		7 [3]	
3	1, 2, 3, 5	5	3	4	14 [5]	7	10 [5]
4	1, 2, 3, 5, 6	6	3	4	12 [6]	7	10
5	1, 2, 3, 5, 6, 4	4	3	4	12	7	10

Ejemplo, el camino mínimo de v_1 a v_6 es 10, la secuencia de vértices que hacen el camino mínimo: $v_1 - v_3 - v_5 - v_6$.