POHON RENTANG (Spanning Tree)

Definition

Let G be a simple graph. A spanning tree of G is a subgraph of G that is a tree containing every vertex of G.

EXAMPLE 1

Find a spanning tree of the simple graph G shown in Figure 2

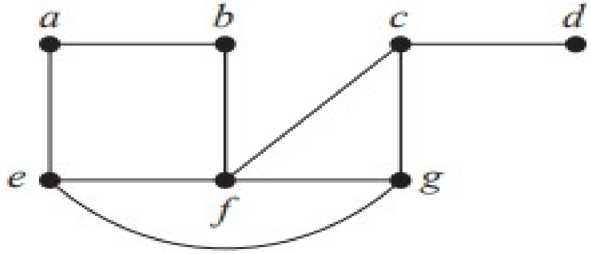
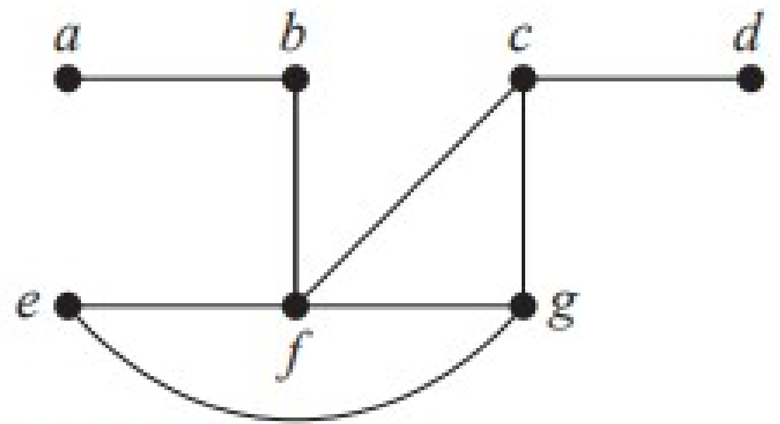


FIGURE 2 The Simple Graph *G*.

Solution:

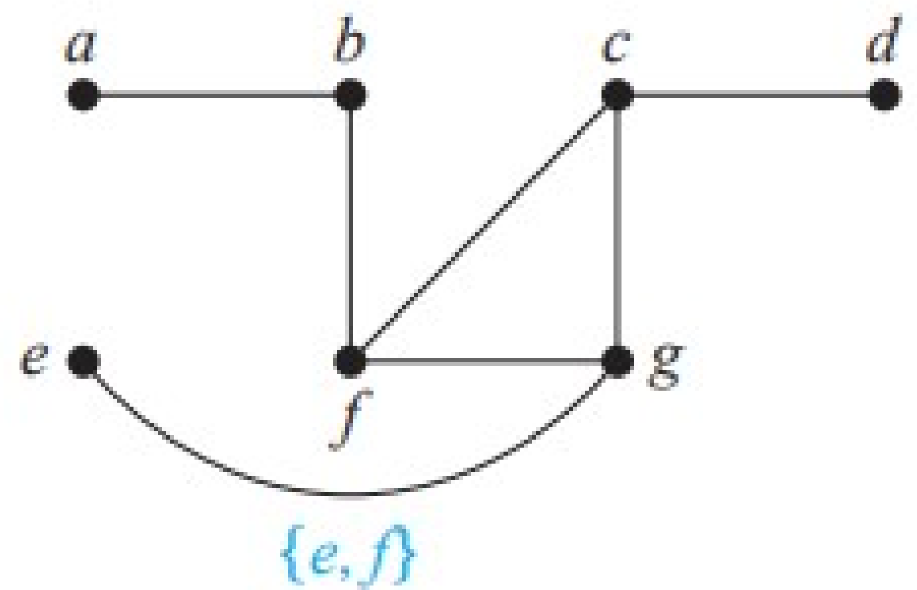
The graph G is connected, but it is not a tree because it contains simple circuits.

Remove the edge {a, e}. This eliminates one simple circuit, and the resulting subgraph is still connected and still contains every vertex of G.

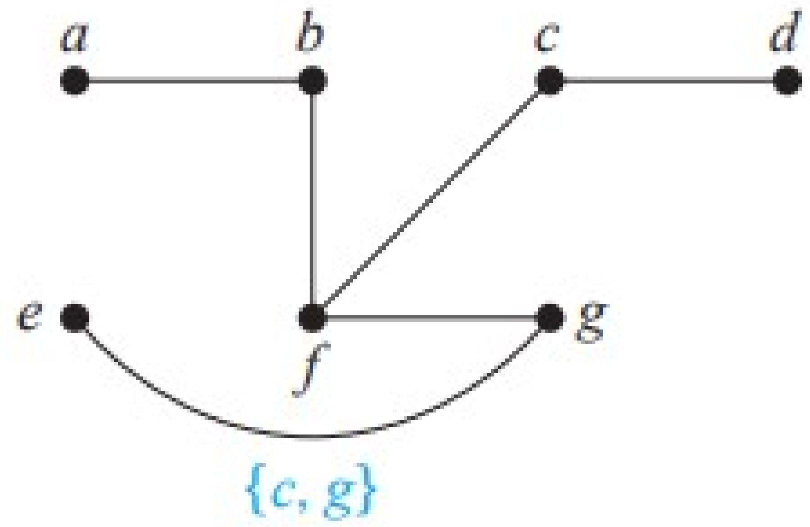


Edge removed: {*a*, *e*}

Next remove the edge {e, f } to eliminate a second simple circuit.



Finally, remove edge {c, g} to produce a simple graph with no simple circuits.



This subgraph is a spanning tree, because it is a tree that contains every vertex of G.

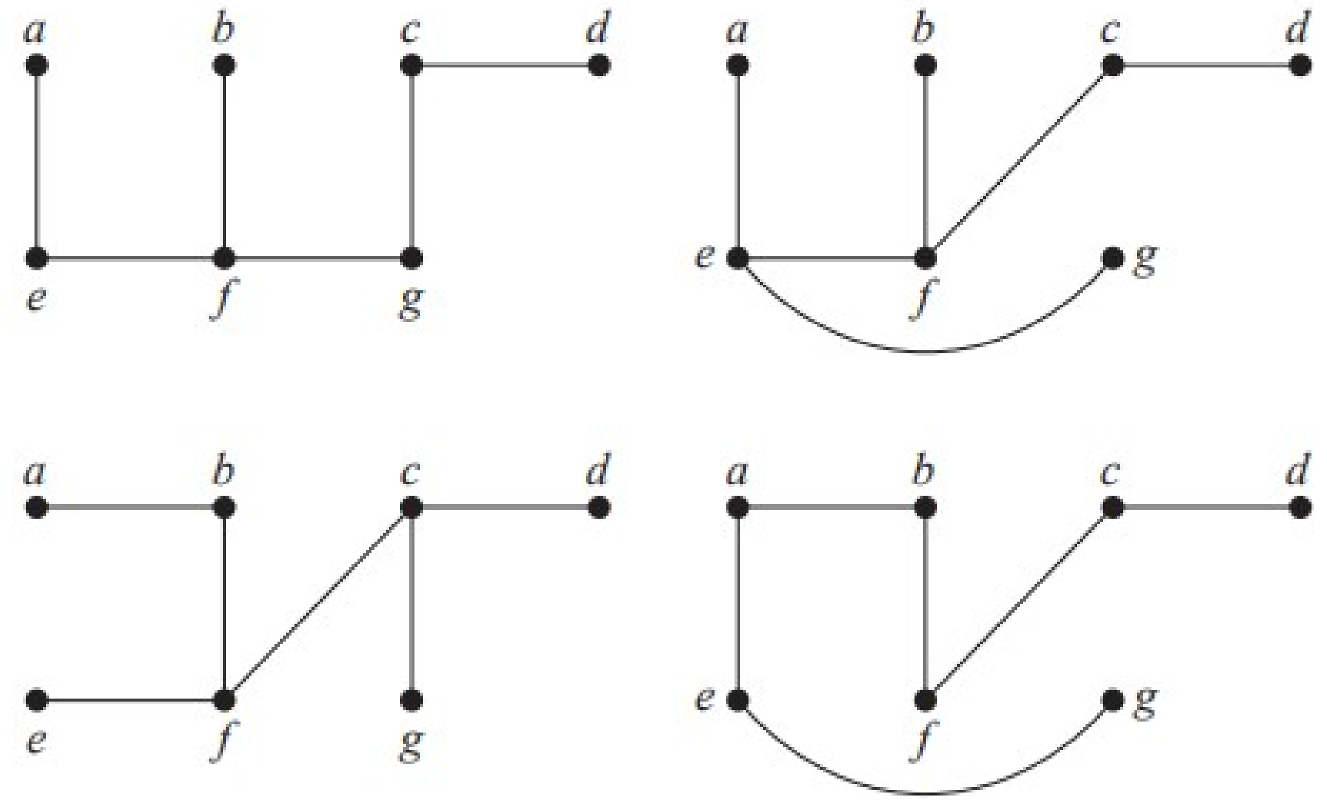
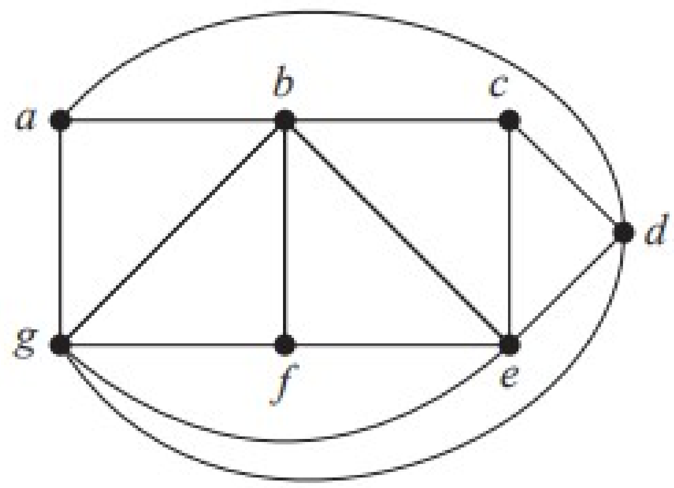
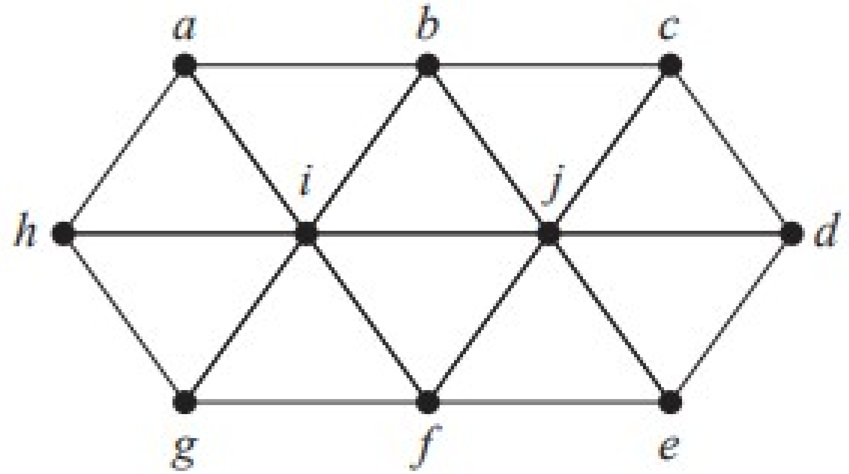


FIGURE 4 **Spanning Trees of *G*.**

Exercises

1. Find a spanning tree for the graph shown by removing edges in simple circuits





2. How many different spanning trees does each of these simple graphs have?

a) K_3 b) K_4 c) $K_{2,2}$ d) C_5

Breadth-First Search

We can also produce a spanning tree of a simple graph by the use of breadth-first search. Again, a rooted tree will be constructed, and the underlying undirected graph of this rooted tree forms the spanning tree. Arbitrarily choose a root from the vertices of the graph. Then add all edges incident to this vertex. The new vertices added at this stage become the vertices at level 1 in the spanning tree. Arbitrarily order them.

Next, for each vertex at level 1, visited in order, add each edge incident to this vertex to the tree as long as it does not produce a simple circuit.

Arbitrarily order the children of each vertex at level 1. This produces the vertices at level 2 in the tree. Follow the same procedure until all the vertices in the tree have been added. The procedure ends because there are only a finite number of edges in the graph. A spanning tree is produced because we have produced a tree containing every vertex of the graph.

Use breadth-first search to find a spanning tree for the graph shown in Figure 9

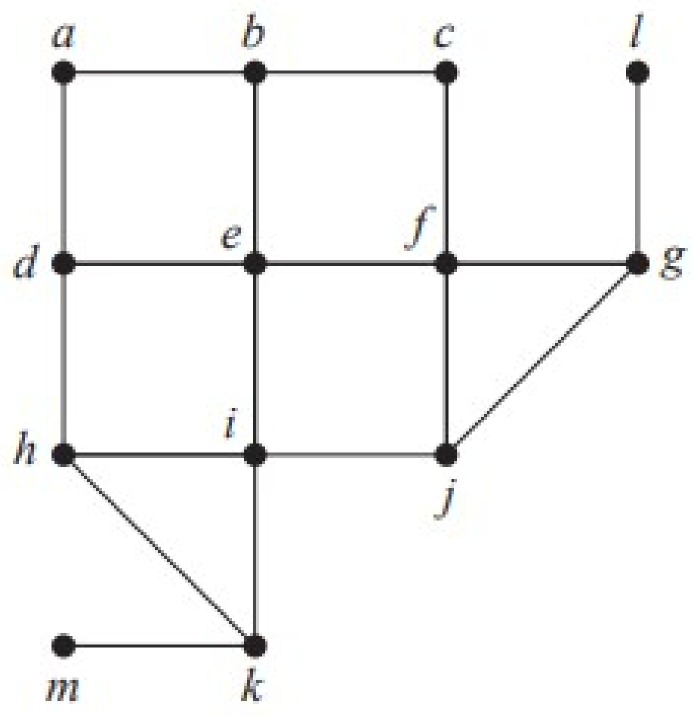
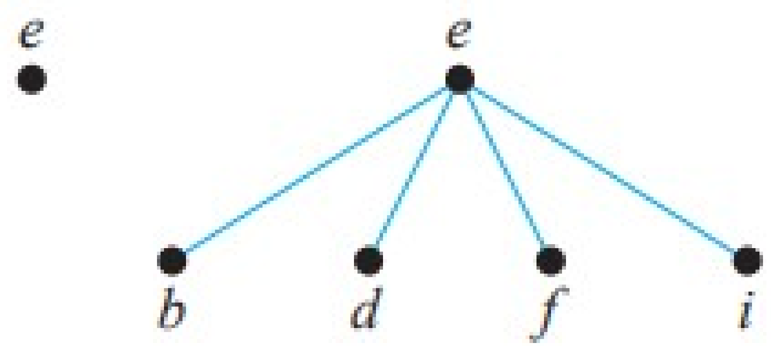


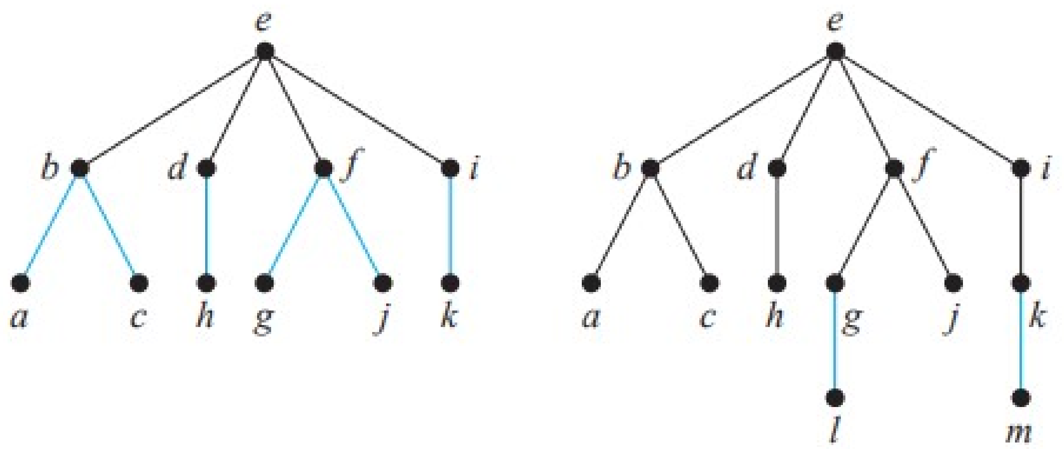
FIGURE 9 **A Graph *G*.**

Solution:

We choose the vertex e to be the root. Then we add edges incident with all vertices adjacent to e, so edges from e to b, d, f , and i are added. These four vertices are at level 1 in the tree.

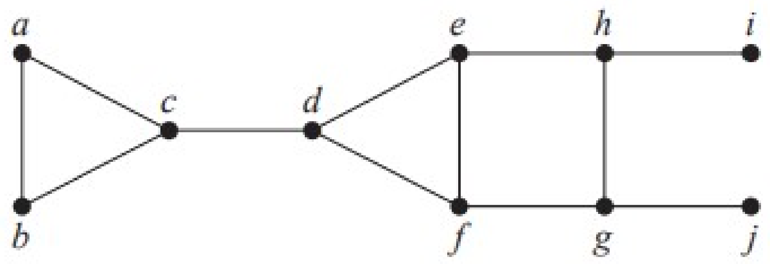


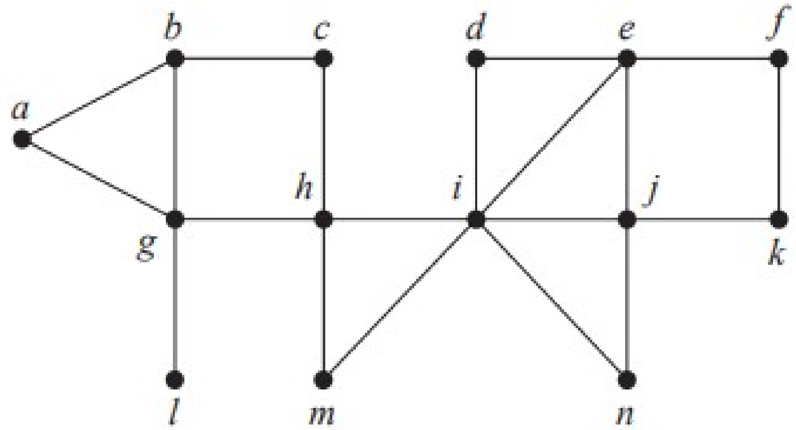
Next, add the edges from these vertices at level 1 to adjacent vertices not already in the tree. Hence, the edges from b to a and c are added, as are edges from d to h, from f to j and g, and from i to k. The new vertices a, c, h, j , g, and k are at level 2. Next, add edges from these vertices to adjacent vertices not already in the graph. This adds edges from g to l and from k to m.



Exercises

Use breadth-first search to produce a spanning tree for each of the simple graphs. Choose a as the root of each spanning tree.

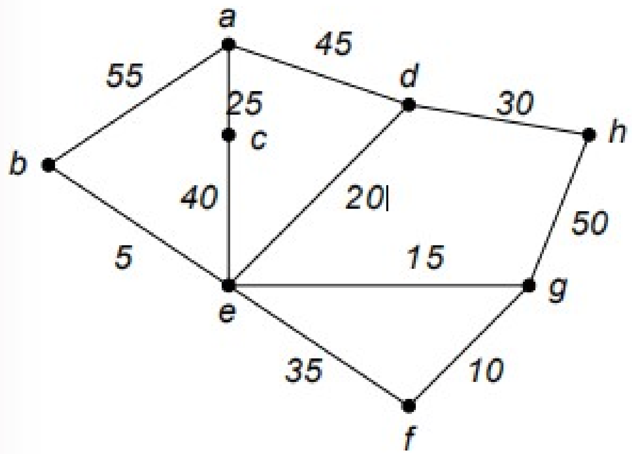


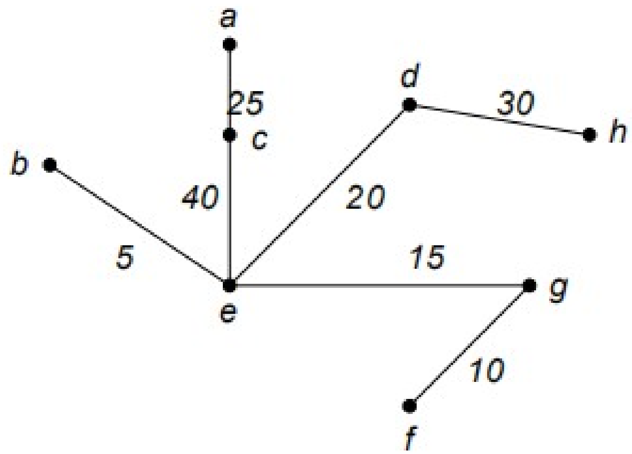


Minimum Spanning Trees

DEFINITION

A minimum spanning tree in a connected weighted graph is a spanning tree that has the smallest possible sum of weights of its edges.





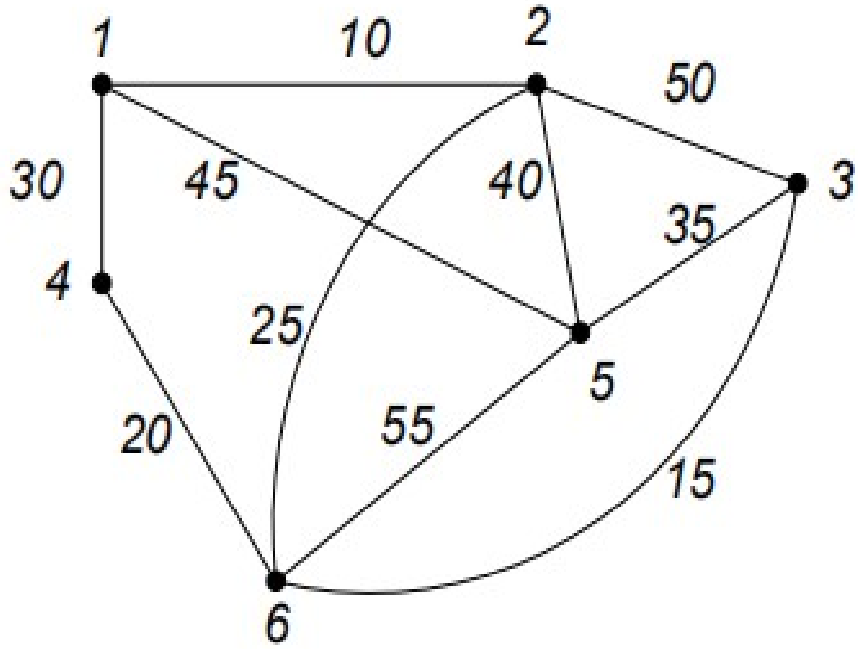
Algoritma Prim

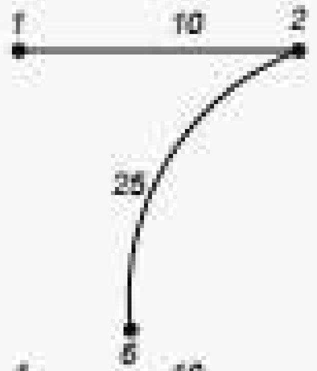
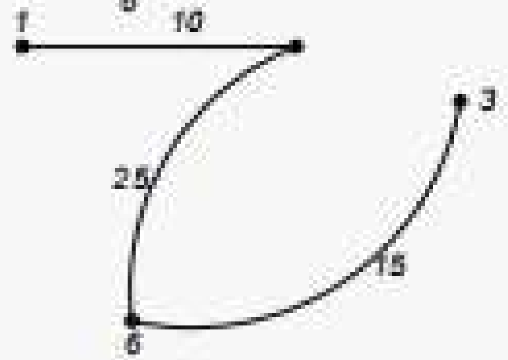
Langkah 1: ambil sisi dari graf G yang berbobot minimum, masukkan ke dalam T.

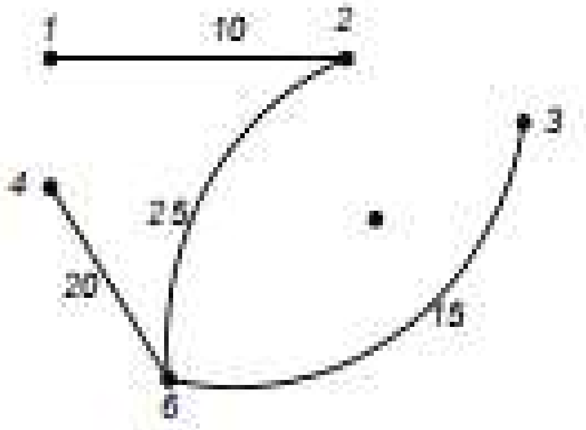
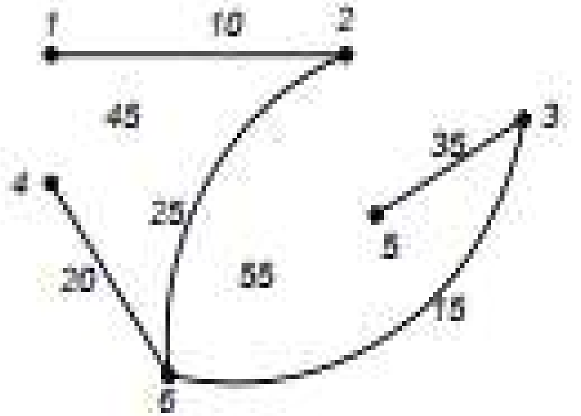
Langkah 2: pilih sisi (u, v) yang mempunyai bobot minimum dan bersisian dengan simpul di T, tetapi (u, v) tidak membentuk sirkuit di T. Masukkan (u, v) ke dalam T.

Langkah 3: ulangi langkah 2 sebanyak $n - 2$ kali.

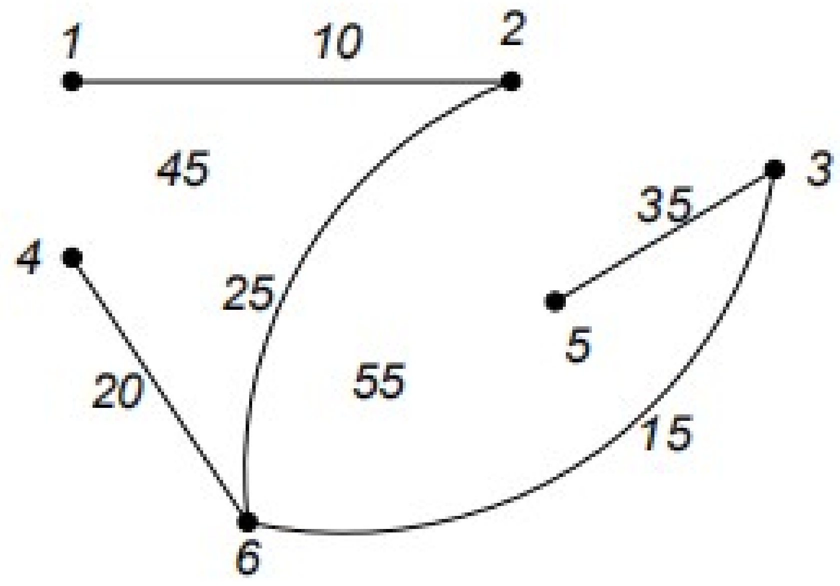
Contoh:



Langkah	Sisi	Bobot	Pohon rentang
1	(1, 2)	10	
2	(2, 6)	25	
3	(3, 6)	15	

4	(4 , 6)	20	
5	(3 , 5)	35	

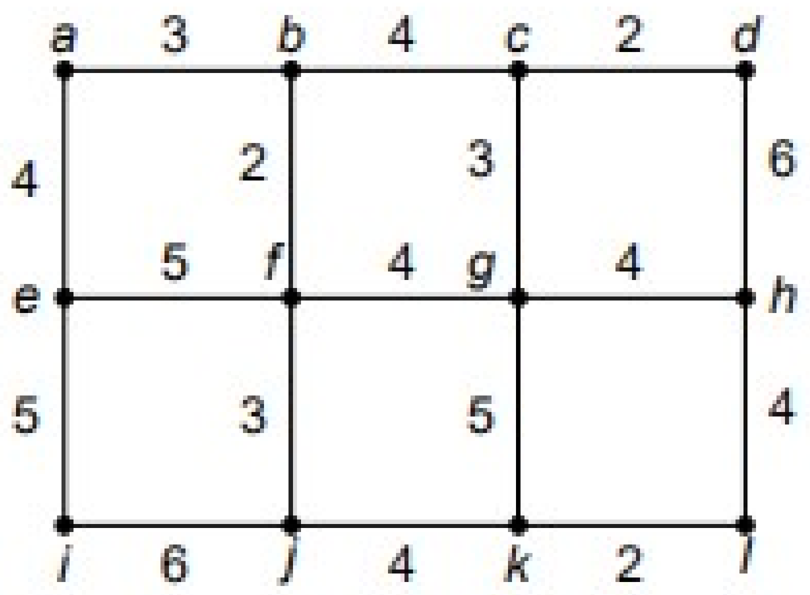
Pohon merentang minimum yang dihasilkan:

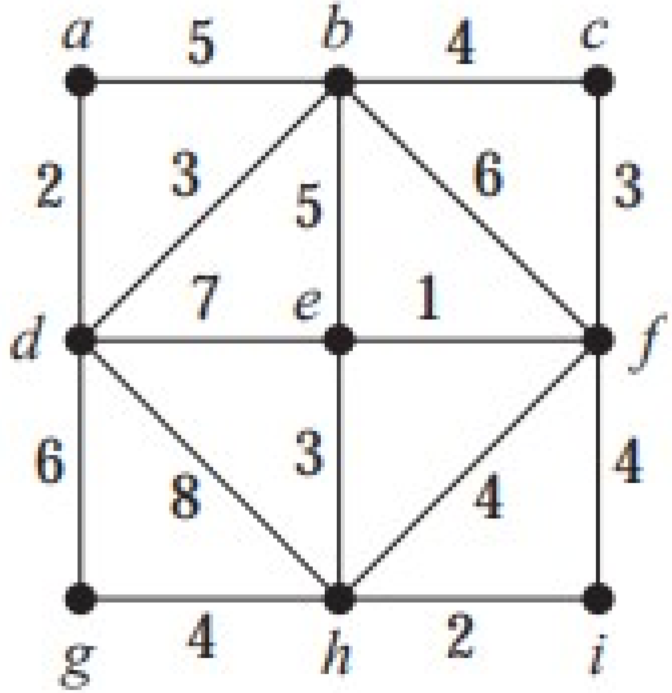


Bobot = 10 + 25 + 15 + 20 + 35 = 105

Latihan

Tentukan pohon merentang minimum dengan menggunakan algoritma prim dari graf berikut:





Algoritma Kruskal

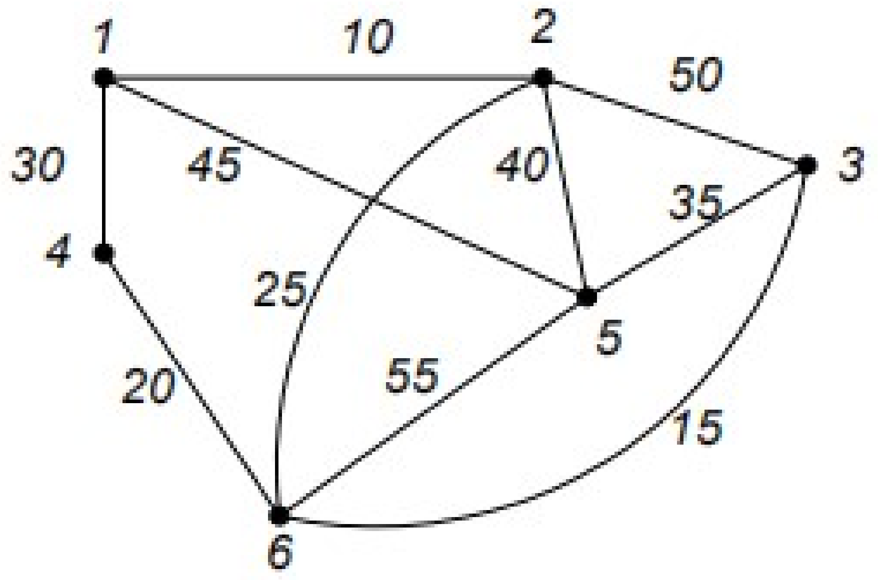
(Langkah 0: sisi-sisi dari graf diurut menaik berdasarkan bobotnya – dari bobot kecil ke bobot besar)

Langkah 1: T masih kosong

Langkah 2: pilih sisi (u, v) dengan bobot minimum yang tidak membentuk sirkuit di T. Tambahkan (u, v) ke dalam T.

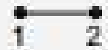
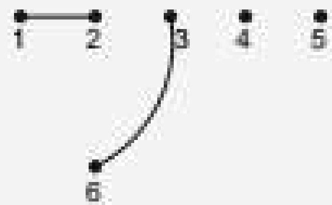
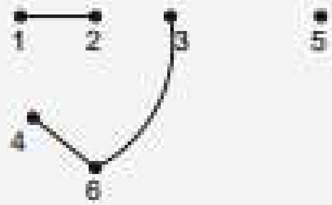
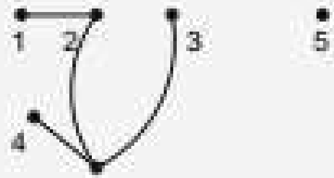
Langkah 3: ulangi langkah 2 sebanyak n – 1 kali

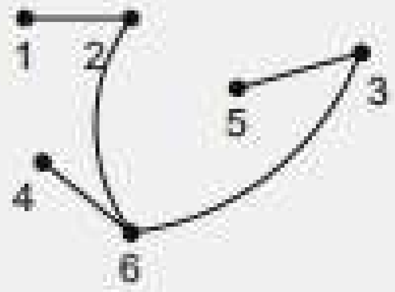
Contoh:



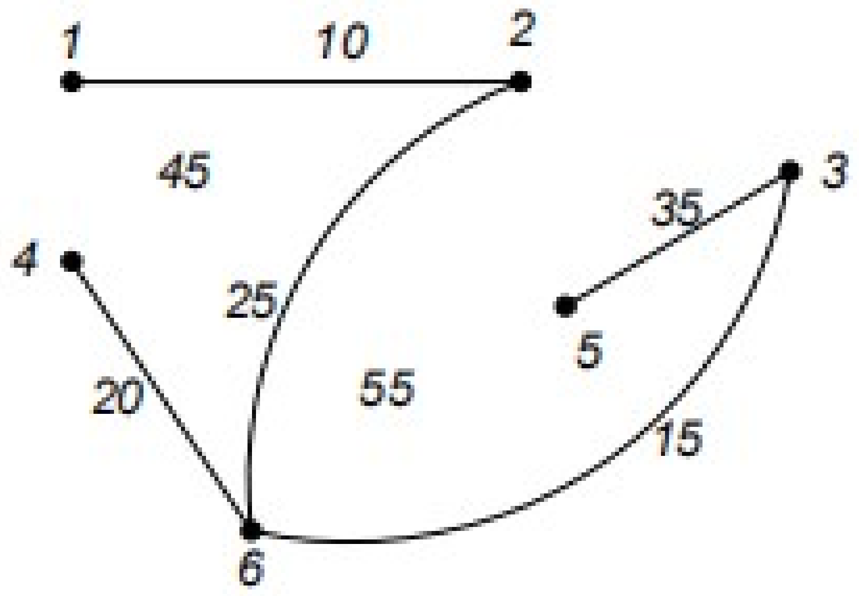
Sisi-sisi diurut menaik:

Sisi	(1,2)	(3,6)	(4,6)	(2,6)	(1,4)	(3,5)	(2,5)	(1,5)	(2,3)	(5,6)
Bobot	10	15	20	25	30	35	40	45	50	55

Langkah	Sisi	Bobot	Hutan merentang
0			
1	(1, 2)	10	
2	(3, 6)	15	
3	(4, 6)	20	
4	(2, 6)	25	

5	(1, 4)	30	ditolak
6	(3, 5)	35	

Pohon merentang minimum yang dihasilkan:



Bobot = 10 + 25 + 15 + 20 + 35 = 105

Latihan

Tentukan pohon merentang minimum dengan menggunakan algoritma kruskal dari graf berikut: