National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Lec # 13,14 & 15

Walk using Matrices

Use for the adjacency matrix is to count the number of walks between two vertices within a graph

Theorem 2.24 Let G be a graph with adjacency matrix A. Then for any integer $n > 0$ the entry a_{ij} in A^n counts the number of walks from v_i to v_j.

Example 1.4.3. Some walks in H of Fig. 1.4.2 and their respective lengths are shown in the table below.

	sequence	walk	length
(1)	$bf_3xf_4wf_5xf_4wf_{10}zf_{10}w$	$b-w$	6
(2)	$bf_3xf_4wf_5xf_7y$	$b-y$	4
(3)	$bf_3xf_4wf_{11}yf_9c$	$b-c$	4
(4)	$bf_2cf_8xf_4wf_5xf_3b$	$b-b$	5
(5)	$bf_2cf_9yf_{12}zf_6xf_3b$	$b-b$	5

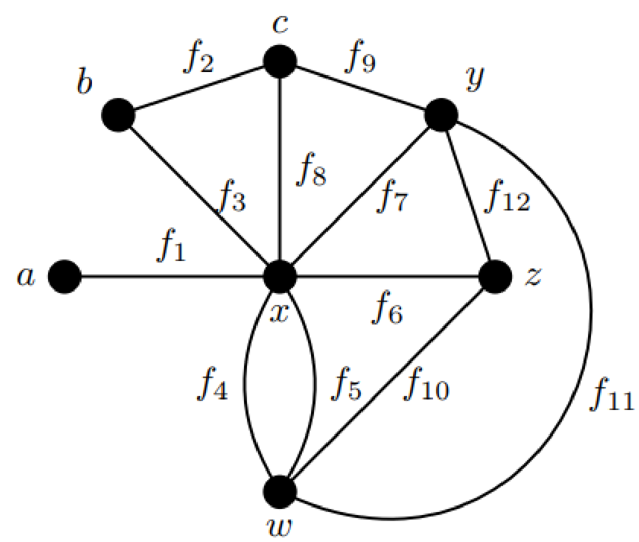


Fig. 1.4.2

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}, \quad A^2 = \begin{bmatrix} 2 & 0 & 2 & 0 & 1 \\ 0 & 3 & 1 & 2 & 1 \\ 2 & 1 & 3 & 0 & 1 \\ 0 & 2 & 0 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 \end{bmatrix}, \quad \text{and} \quad A^4 = \begin{bmatrix} 9 & 3 & 11 & 1 & 6 \\ 3 & 15 & 7 & 11 & 8 \\ 11 & 6 & 15 & 3 & 8 \\ 1 & 11 & 3 & 9 & 6 \\ 6 & 8 & 8 & 6 & 8 \end{bmatrix}$$

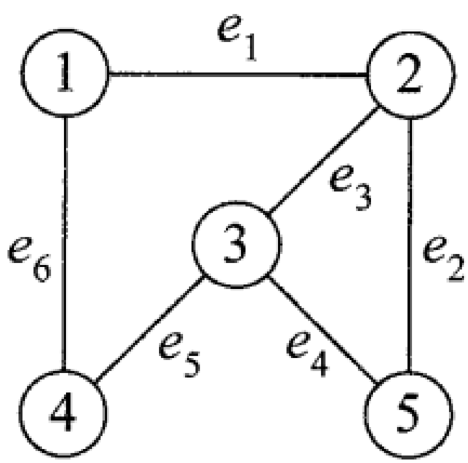
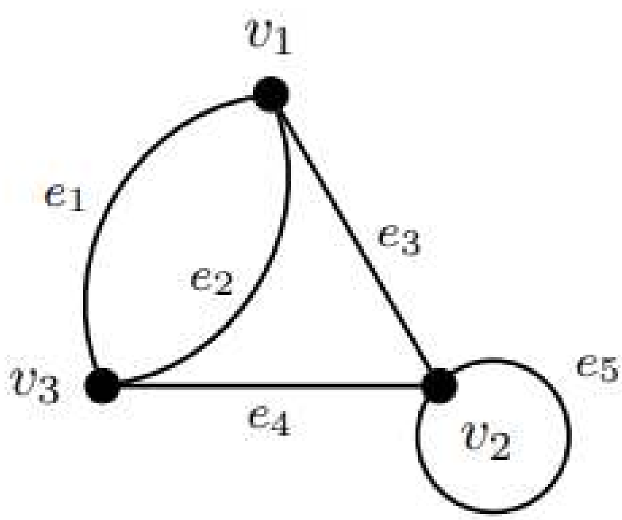
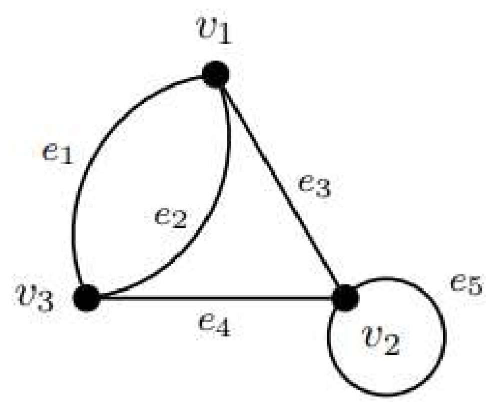


Fig. 2-2

The degrees of the vertices 1, 2, 3, 4, 5 are 2, 3, 3, 2, 2, respectively, in agreement with the diagonal entries of A^2. The (1, 5) entry in A^4 is 6, indicating that there are six *different* walks of length 4 between 1 and 5. These walks are $1-4-1-2-5$, $1-2-1-2-5$, $1-4-3-2-5$, $1-2-5-2-5$, $1-2-3-2-5$, and $1-2-5-3-5$.



$$\begin{bmatrix} 0 & 1 & 2 \\ 1 & 1 & 1 \\ 2 & 1 & 0 \end{bmatrix}$$



$$\begin{bmatrix} 0 & 1 & 2 \\ 1 & 1 & 1 \\ 2 & 1 & 0 \end{bmatrix}$$

Now consider the walks from v_1 to v_2. There is only one walk of length 1, and yet three of length 2:

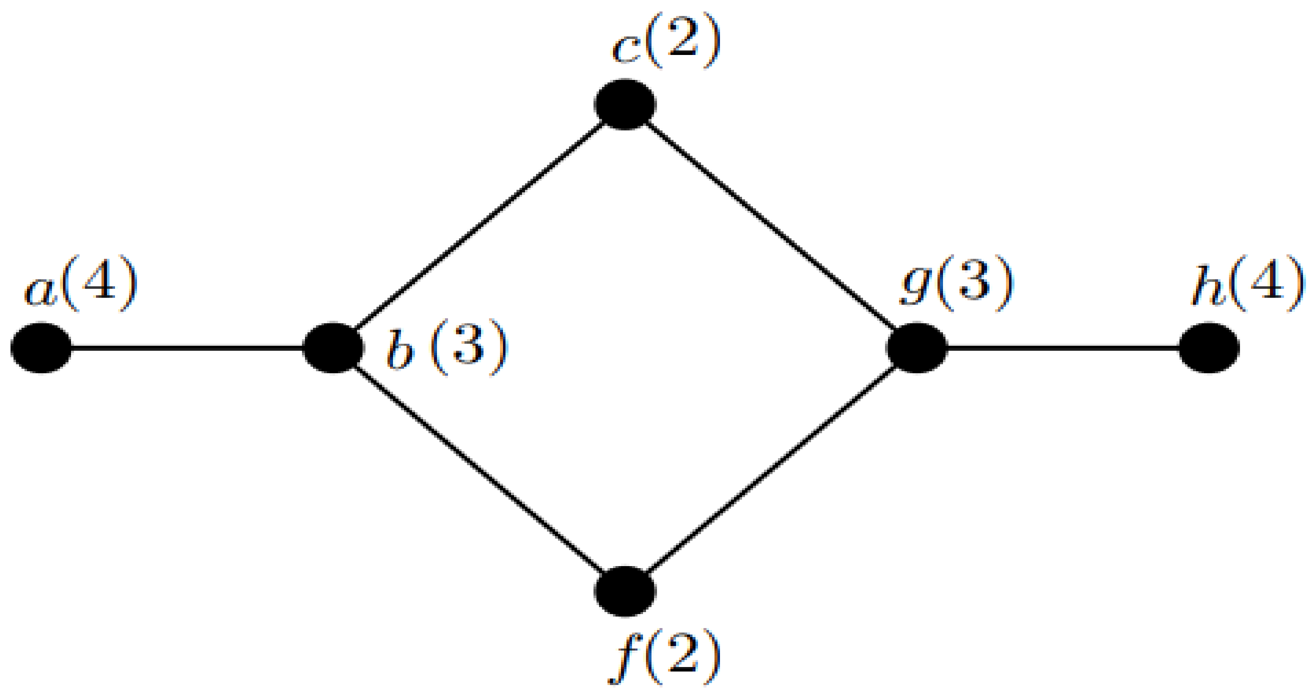
$$v_1 \underset{e_3}{\rightarrow} v_2 \underset{e_5}{\rightarrow} v_2$$
$$v_1 \underset{e_1}{\rightarrow} v_3 \underset{e_4}{\rightarrow} v_2$$
$$v_1 \underset{e_2}{\rightarrow} v_3 \underset{e_4}{\rightarrow} v_2$$

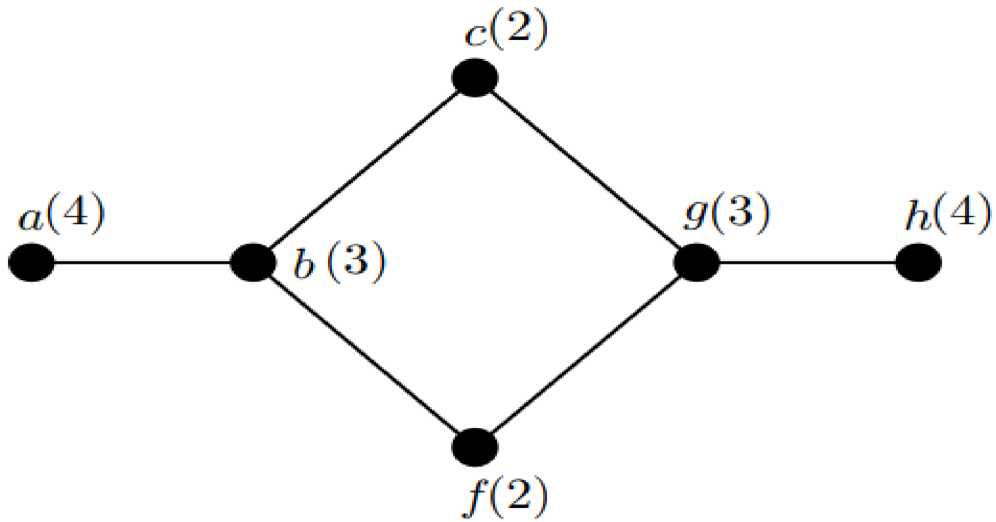
$$A^2 = \begin{bmatrix} 5 & 3 & 1 \\ 3 & 3 & 3 \\ 1 & 3 & 5 \end{bmatrix}$$

Definition 2.25 Given two vertices x, y *in* a graph G, define the ***distance*** $d(x, y)$ as the length of the shortest path from x to y. The ***eccentricity*** of a vertex x is the maximum distance from x to any other vertex in G; that is $\epsilon(x) = \max_{y \in V(G)} d(x, y)$.

The ***diameter*** of G is the maximum eccentricity among all vertices, and so measures the maximum distance between any two vertices; that is $diam(G) = \max_{x,y \in V(G)} d(x, y)$. The ***radius*** of a graph is the minimum eccentricity among all vertices; that is $rad(G) = \min_{x \in V(G)} \epsilon(x)$.

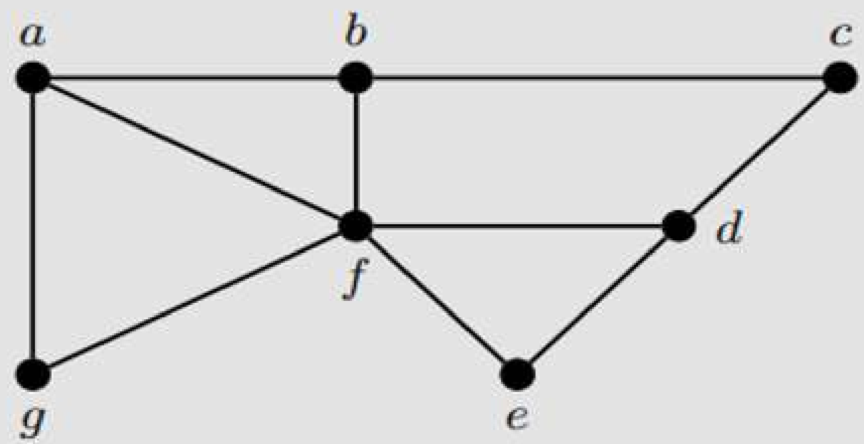
Radius & Diameter?





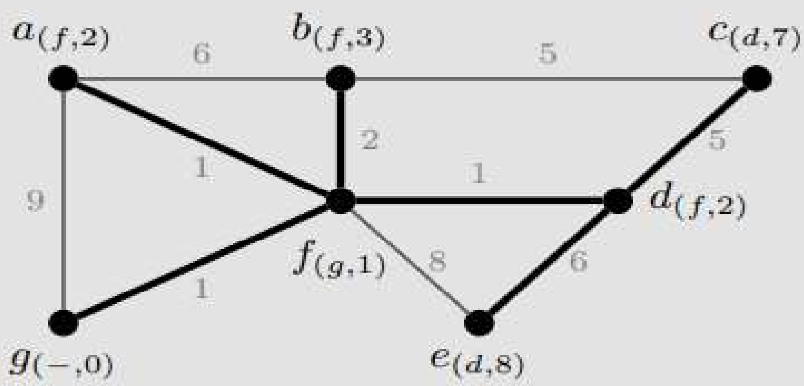
Among the six eccentricities shown in the figure, we notice that '2' is the smallest while '4' is the largest. In this situation, we say that the **radius** of G is 2 and the **diameter** of G is 4. Note also that there are two vertices in G, namely c and f, with least eccentricity (i.e., $e(c) = e(f) = 2$). Each of them is called a **central** vertex, and the set of these two central vertices is called the **center** of G.

Example 2.18 Find the diameter and radius for the graph below.



Remember!

Step 7: Let $u = c$. Then the neighbors of all highlighted vertices are $F = \{e\}$. However, we do not need to update any labels since c and e are not adjacent. Thus we highlight the edge de and the vertex e. This terminates the iterations of the algorithm since all vertices are now highlighted.

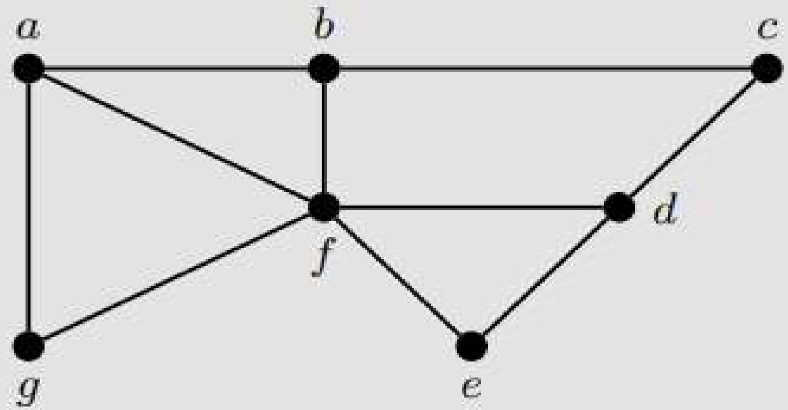


$$F = \{e\}$$

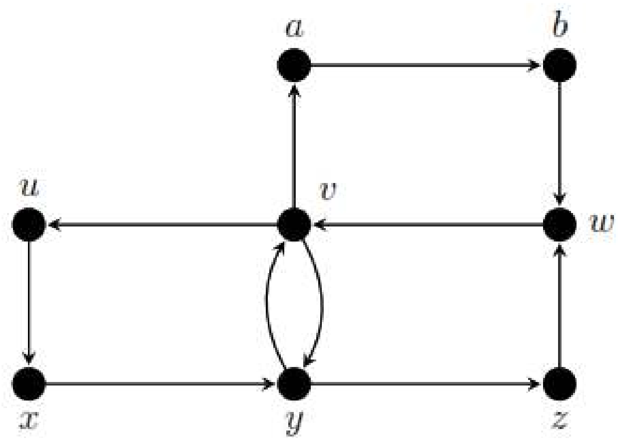
$$\begin{array}{c|c} a & (f,2) \\ b & (f,3) \\ c & (d,7) \\ d & (f,2) \\ e & (d,8) \\ f & (g,1) \\ g & (-,0) \end{array}$$

Output: The shortest paths from g to all other vertices can be found highlighted above. For example the shortest path from g to c is $g\,f\,d\,c$ and has a total weight 7, as shown by the label of c.

Example 2.18 Find the diameter and radius for the graph below.



Solution: Note that f is adjacent to all vertices except c, but there is a path of length 2 from f to c. As no vertex is adjacent to all other vertices, we know the radius is 2. The longest path between two vertices is from g to c, and is of length 3, so the diameter is 3.



Example 10.2.6. For the digraph D in Example 10.2.4 (see Fig. 10.2.3), we have:

$$\begin{aligned} e(z) &= \max\{d(z,a), d(z,b), d(z,u), d(z,v), d(z,w), d(z,x), d(z,y), d(z,z)\} \\ &= \max\{3,4,3,2,1,4,3,0\} = 4. \end{aligned}$$

It can be verified that the eccentricities of the eight vertices in D are shown in the following table:

Vertex	a	b	u	v	w	x	y	z
Eccentricity	5	4	5	3	3	4	3	4

Example 10.2.6. For the digraph D in Example 10.2.4 (see Fig. 10.2.3), we have:

$$\begin{aligned} e(z) &= \max\{d(z,a), d(z,b), d(z,u), d(z,v), d(z,w), d(z,x), d(z,y), d(z,z)\} \\ &= \max\{3,4,3,2,1,4,3,0\} = 4. \end{aligned}$$

It can be verified that the eccentricities of the eight vertices in D are shown in the following table:

Vertex	a	b	u	v	w	x	y	z
Eccentricity	5	4	5	3	3	4	3	4

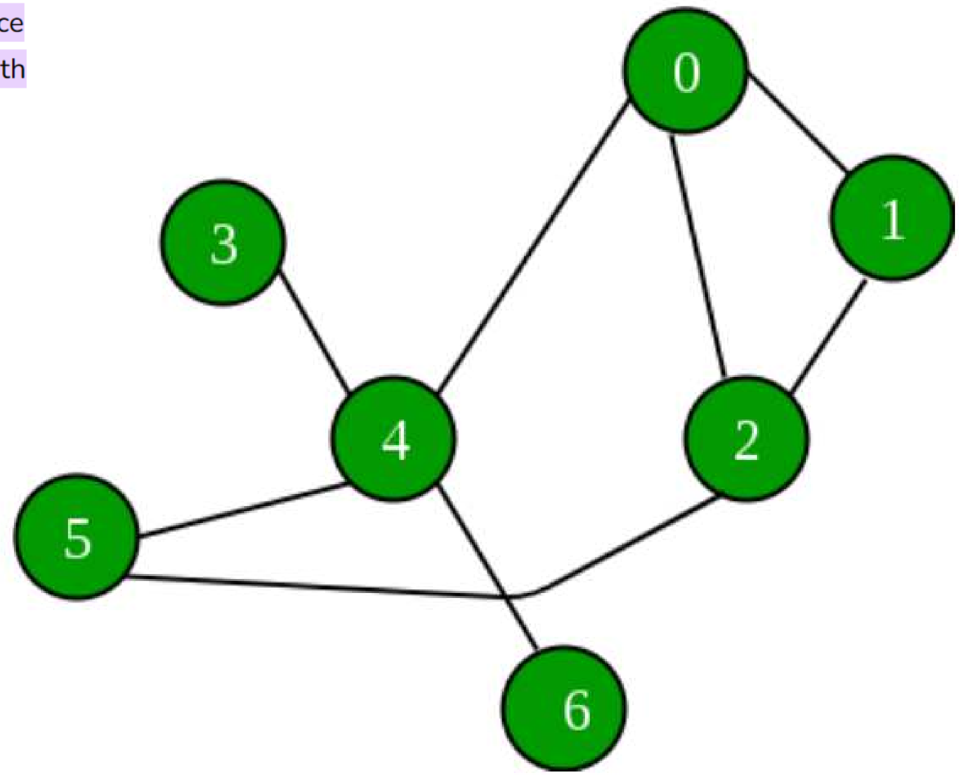
For the digraph D in Example 10.2.4,

$$\operatorname{diam}(D) = 5 = d(a,x) = d(a,z) = d(u,b) = \max\{d(p,q) \mid p,q \in V(D)\}.$$

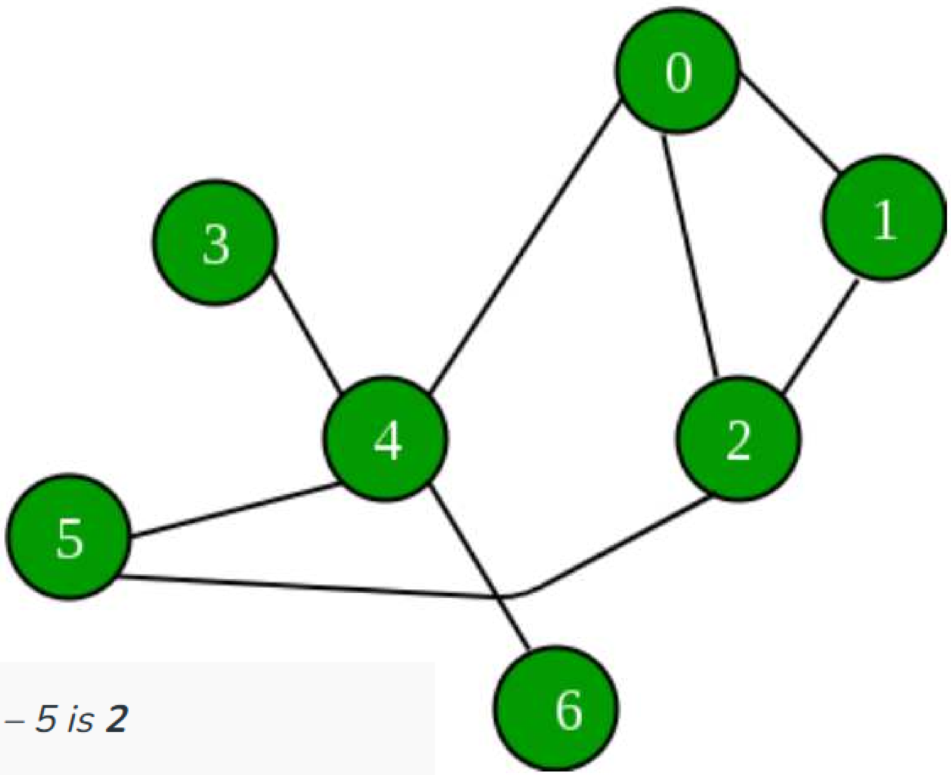
Distance from 1-5

2. The distance between two Vertices –

The distance between two vertices in a graph is the number of edges in a shortest or minimal path. It gives the available minimum distance between two edges. There can exist more than one shortest path between two vertices.



Distance from 1-5

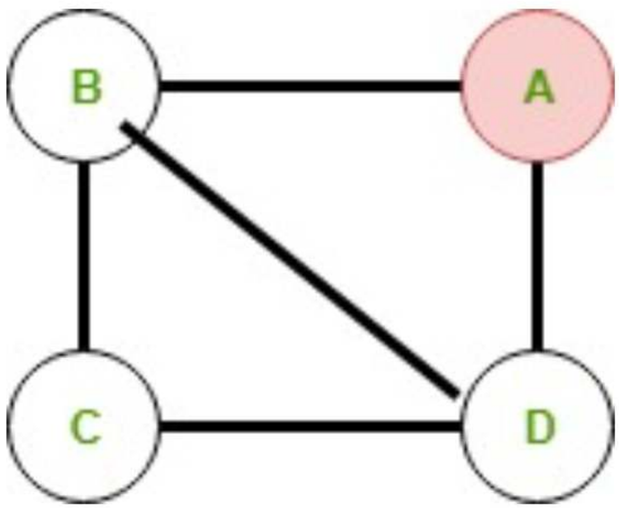


*Shortest Distance between 1 – 5 is **2***

1 → 2 → 5

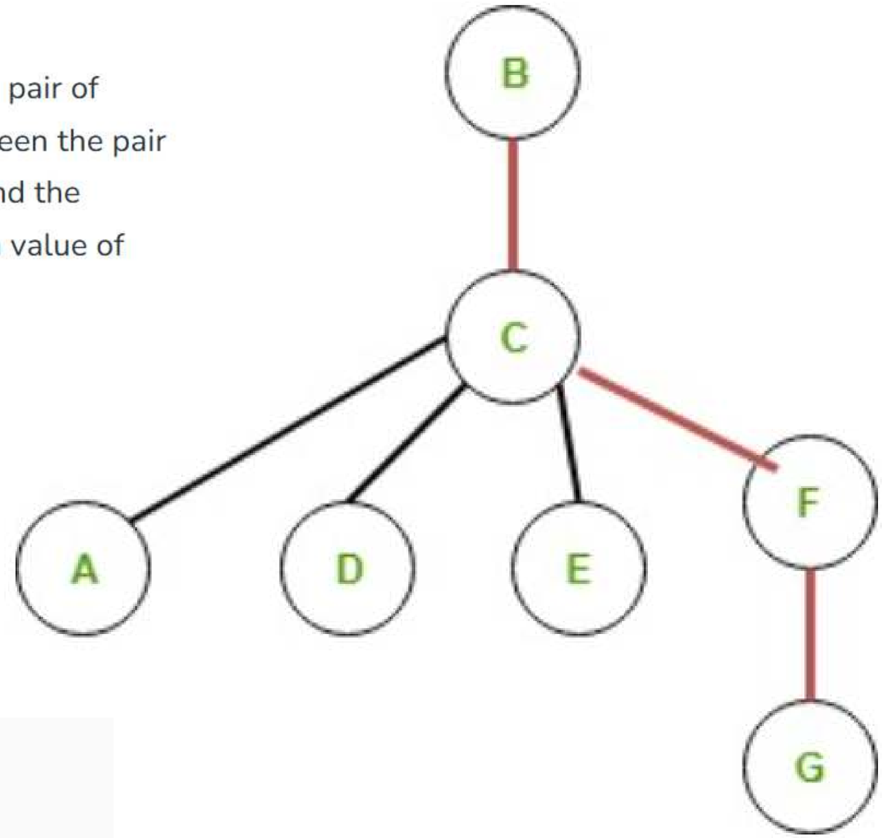
3. Eccentricity of graph –

It is defined as the maximum distance of one vertex from other vertex. The maximum distance between a vertex to all other vertices is considered as the eccentricity of the vertex. It is denoted by e(V).



4. Diameter of graph –

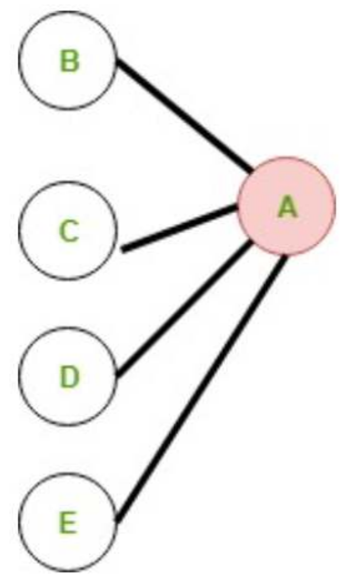
The diameter of graph is the maximum distance between the pair of vertices. It can also be defined as the maximal distance between the pair of vertices. Way to solve it is to find all the paths and then find the maximum of all. It can also be found by finding the maximum value of eccentricity from all the vertices.



Diameter: ***3***
BC → CF → FG
Here the eccentricity of the vertex B is 3 since (B,G) = 3. (Maximum Eccentricity of Graph)

6. Centre of graph –

It consists of all the vertices whose eccentricity is minimum. Here the eccentricity is equal to the radius. For example, if the school is at the center of town it will reduce the distance buses has to travel. If eccentricity of two vertex is same and minimum among all other both of them can be the center of the graph.

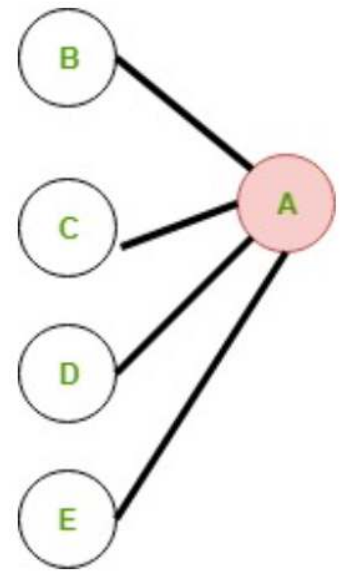


*Centre: **A***

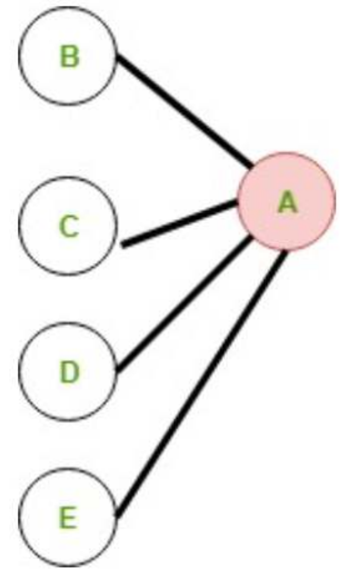
Inorder to find the center of the graph, we need to find the eccentricity of each vertex and find the minimum among all of them.

The minimum eccentricity vertex will be considered as the center.

Definition 2.29 Let G be a graph with $rad(G) = r$. Then x is a ***central vertex*** if $\epsilon(x) = r$. Moreover, the ***center*** of G is the graph $C(G)$ that is induced by the central vertices of G.



Definition 2.29 Let G be a graph with $rad(G) = r$. Then x is a ***central vertex*** if $\epsilon(x) = r$. Moreover, the ***center*** of G is the graph $C(G)$ that is induced by the central vertices of G.



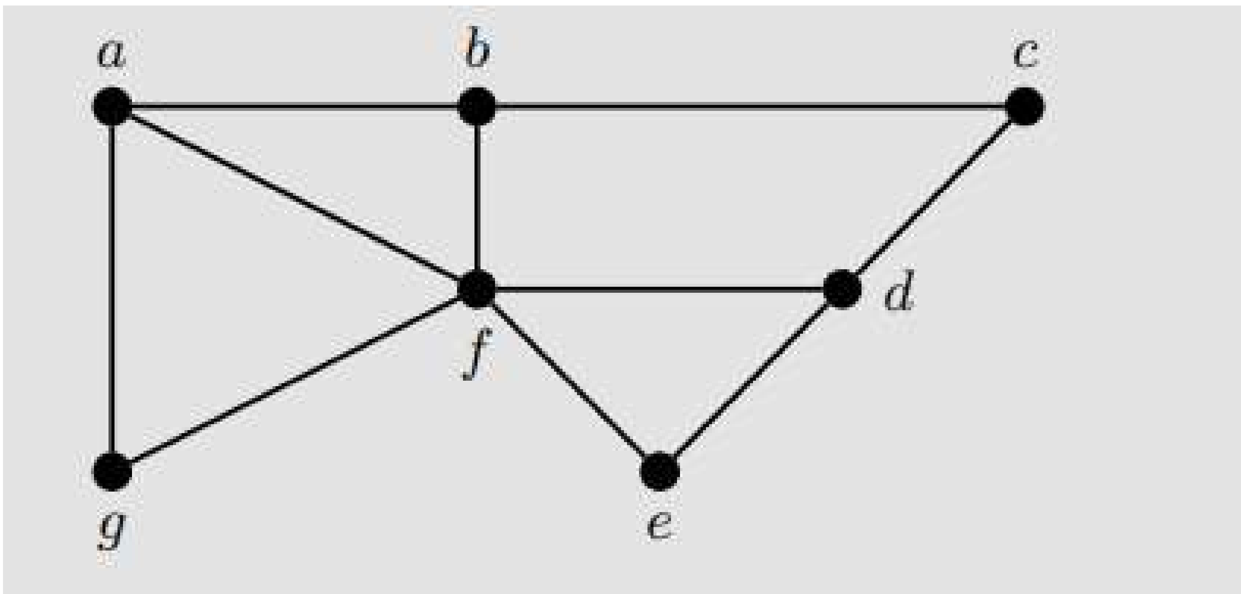
Theorem 2.26 If G is disconnected then $\overline{G}$ is connected and $diam(\overline{G}) \leq 2$.

Theorem 2.27 For a simple graph G if $rad(G) \geq 3$ then $rad(\overline{G}) \leq 2$.

Theorem 2.28 For any simple graph G, $rad(G) \leq diam(G) \leq 2rad(G)$.

Definition 2.30 Given a graph G, the ***girth*** of G, denoted $g(G)$, is the minimum length of a cycle in G. The ***circumference*** of G is the maximum length of a cycle.

Example 2.20 Find the girth and circumference for the graph from Example 2.18.



Example 2.20 Find the girth and circumference for the graph from Example 2.18.

Solution: Since the graph is simple, we know the girth must be at least 3, and since we can find triangles within the graph we know $g(G) = 3$. Moreover, the circumference is 7 since we can find a cycle containing all the vertices (try it!).

EX, #:2.4

Problems: 2.1-2.9, 2.15, 2.16, 2.27, 2.28

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Tree

Definition 3.1 A graph G is

- ***acyclic*** if there are no cycles or circuits in the graph.
- a ***tree*** if it is both acyclic and connected.
- a ***forest*** if it is an acyclic graph.

In addition, a vertex of degree 1 is called a ***leaf***.

- A tree is a connected graph without cycles

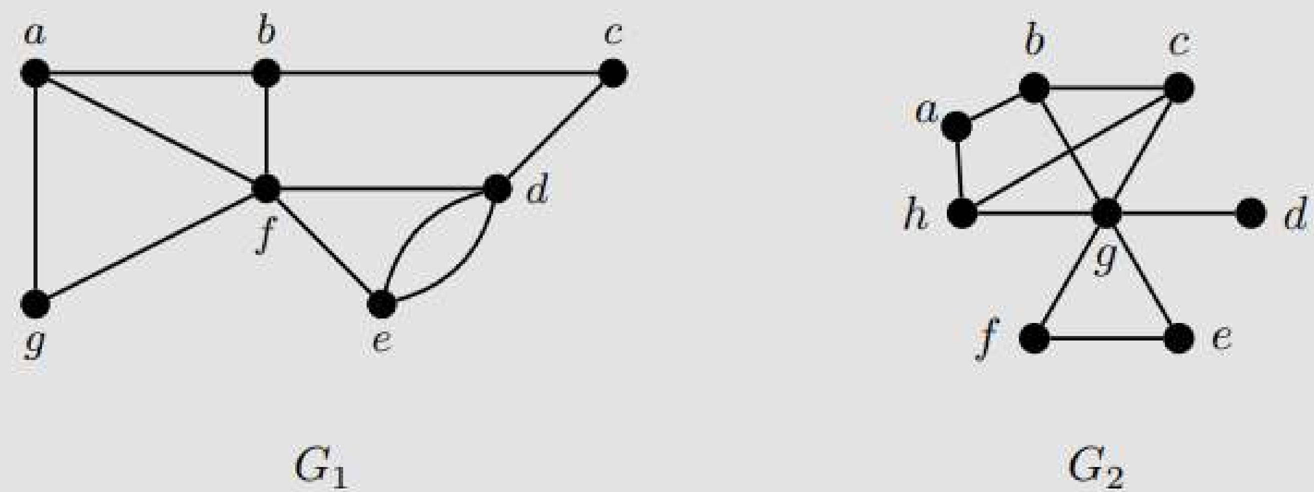
- A tree is a connected graph on n vertices with $n - 1$ edges

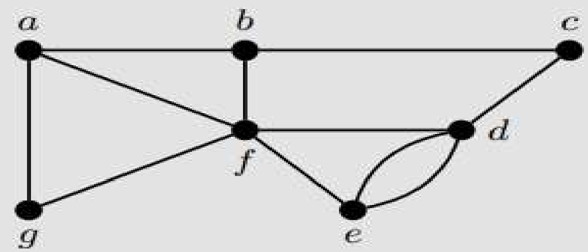
- A graph is a tree if and only if there is a unique simple path between any pair of its vertices

Definition 3.2 A ***spanning tree*** is a spanning subgraph that is also a tree.

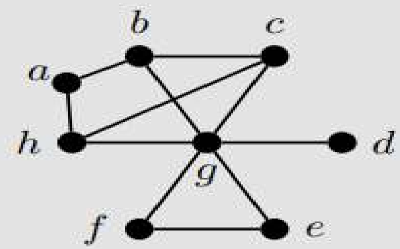
- A Spanning Tree of a graph *G*, is a subgraph of *G* which is a tree and contains all vertices of *G*
- A Minimum Spanning Tree of a weighted graph *G* is a spanning tree of the smallest weight

Example 3.1 For each of the graphs below, find a spanning tree and a subgraph that does not span.



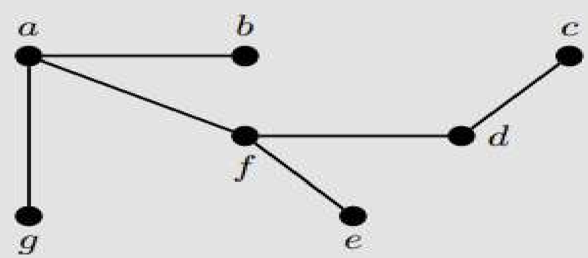


G_1

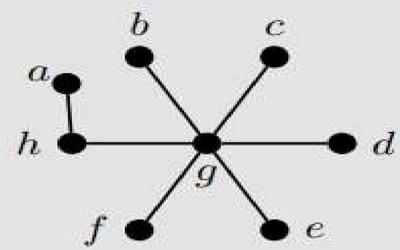


G_2

Solution: To find a spanning tree, we must form a subgraph that is connected, acyclic, and includes every vertex from the original graph. The graphs T_1 and T_2 below are two examples of spanning trees for their respective graphs; other examples exist.

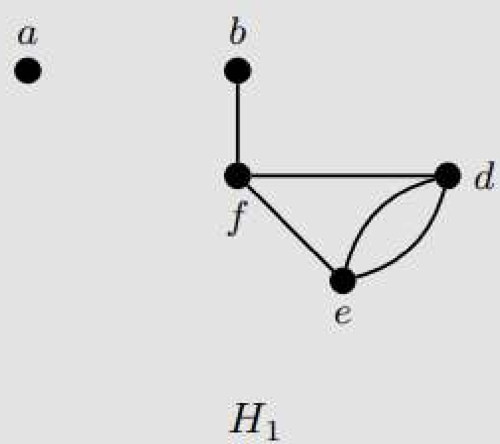


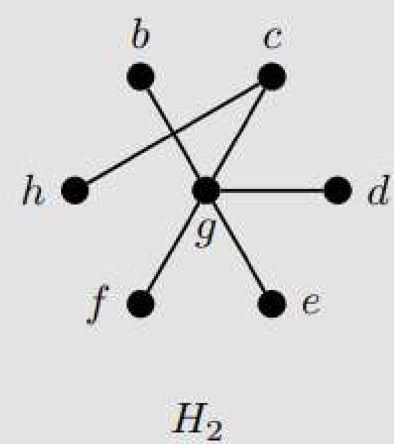
T_1



T_2

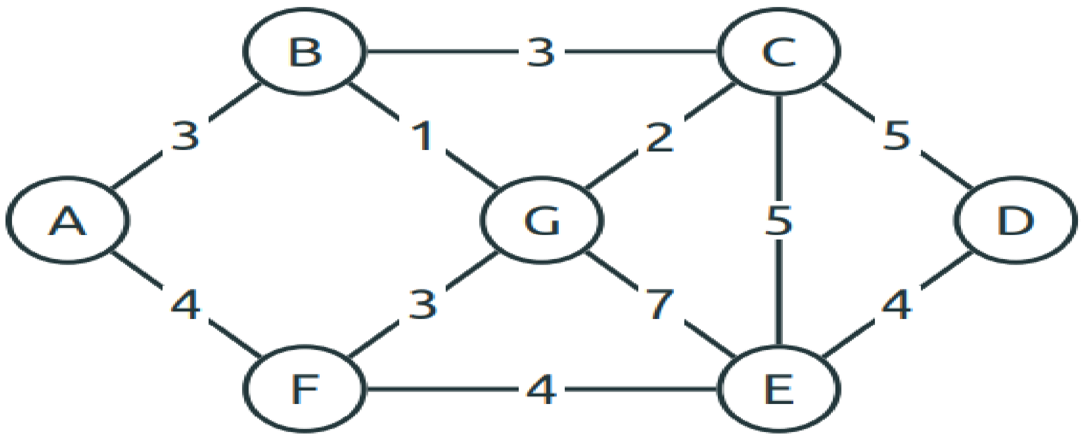
The subgraph H_1 below is neither spanning nor a tree since some vertices from G_1 are missing and there is a multi-edge (and hence a circuit) between d and e. The subgraph H_2 below is not spanning since it does not contain vertex a, but it is a tree since no circuits or cycles exist. As above, these are merely examples and other non-spanning subgraphs exist.





Definition 3.3 Given a weighted graph $G = (V, E, w)$, T is a ***minimum spanning tree***, or MST, of G if it is a spanning tree with the least total weight.

Minimum Spanning Tree: Examples



Kruskal's Algorithm

Input: Weighted connected graph $G = (V, E)$.

Steps:

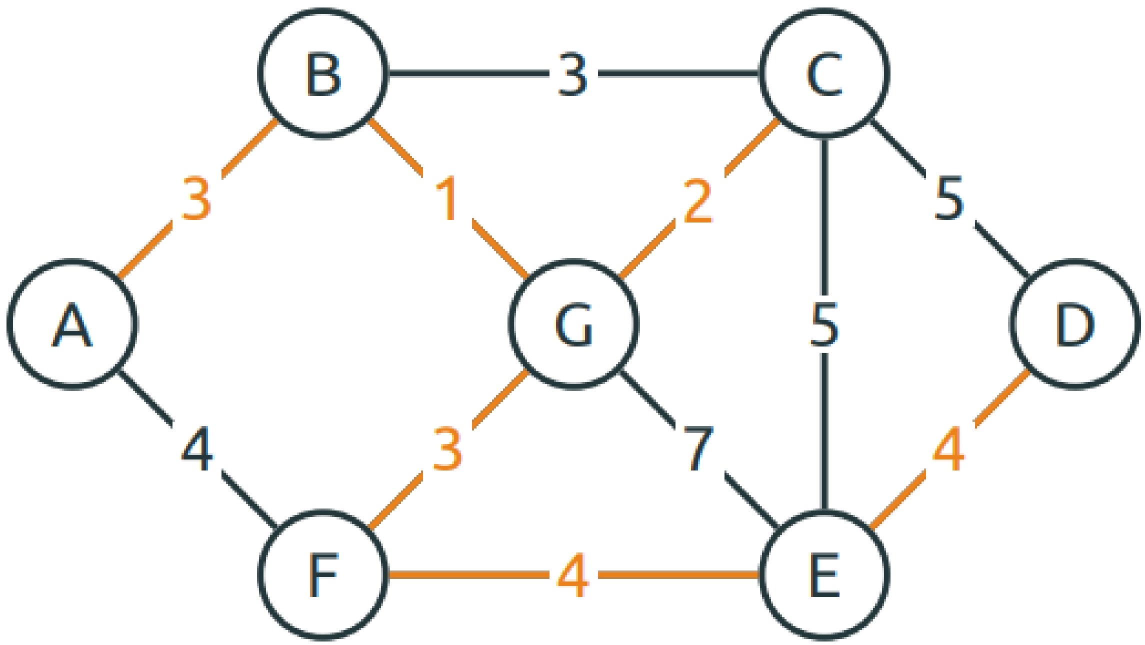
1. Choose the edge of least weight. Highlight it and add it to $T = (V, E')$.
2. Repeat Step (1) so long as no circuit is created. That is, keep picking the edges of least weight but skip over any that would create a cycle in T.

Output: Minimum spanning tree T of G.

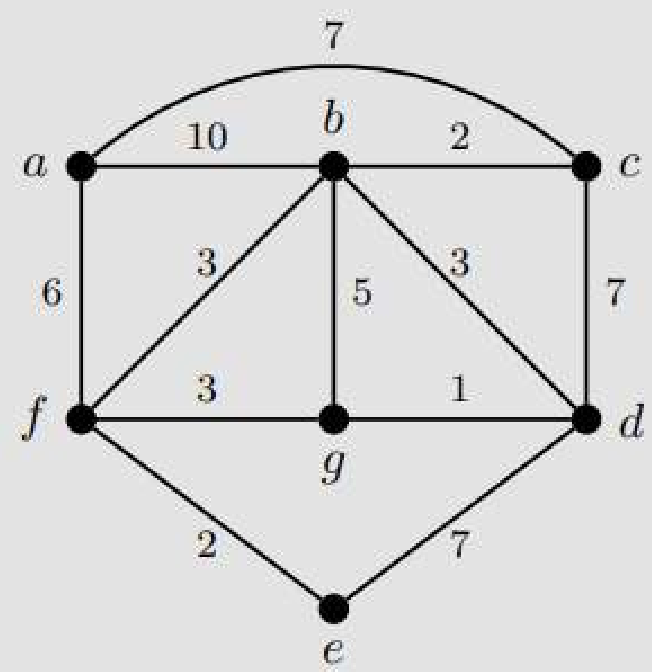
National University of computer and emerging sciences

Foundation for Advancement of Science and Technology FAST

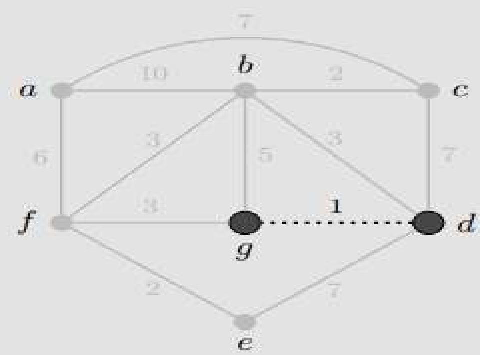
Minimum Spanning Tree: Examples



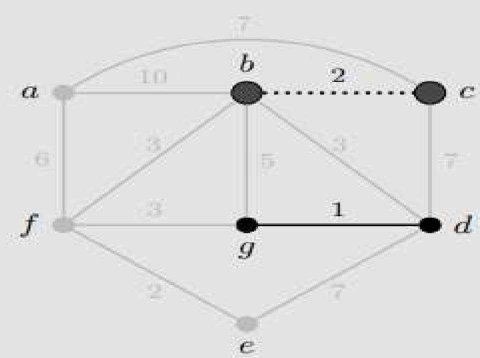
Example 3.2 Find the minimum spanning tree of the graph G below using Kruskal's Algorithm.

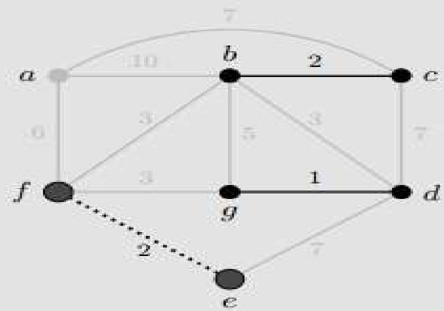


Step 1: Pick the smallest edge, gd and highlight it.

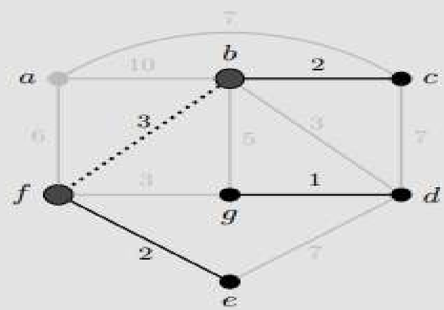


Step 2: Pick the next smallest edge. There are two edges of weight 2 (bc and ef). Either is a valid choice. We choose bc.

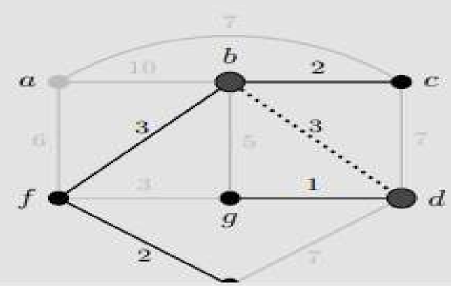


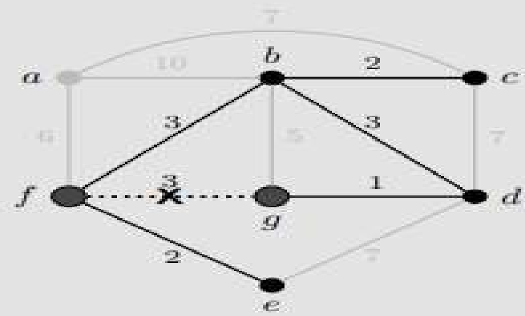


Step 4: The next smallest edge weight is 3, and there are 3 edges to choose from (bd, bf, and fg) . We randomly pick bf.

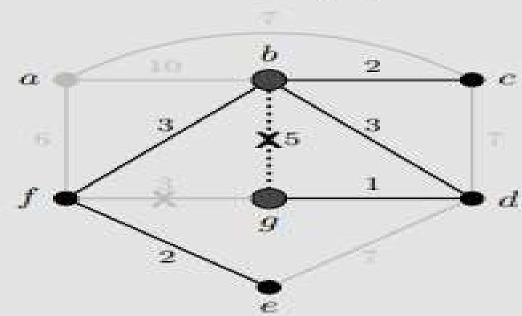


Step 5: Both of the other edges of weight 3 are still available. We choose bd.

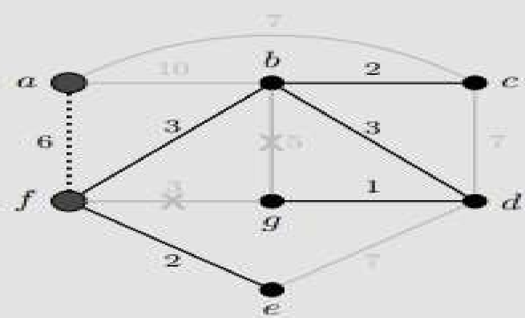


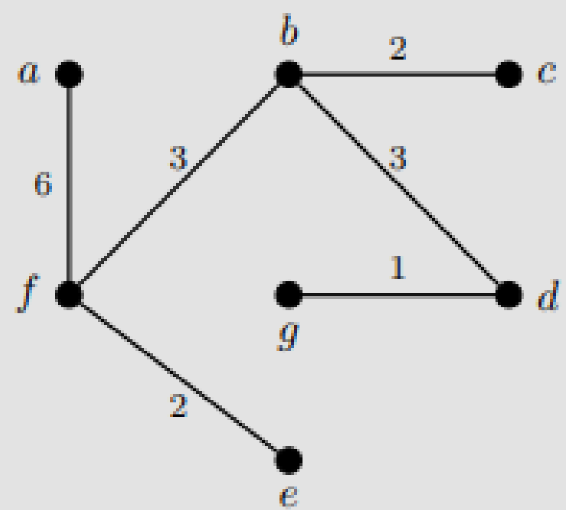


The next smallest edge is bg of weight 5. Again, we cannot choose this edge since it would create a circuit $(b\,d\,g\,b)$.



The next available edge is af of weight 6. This is also the last edge needed since we now have a tree containing all the vertices of G.





Output: The following tree with total weight 17.

Prim's Algorithm

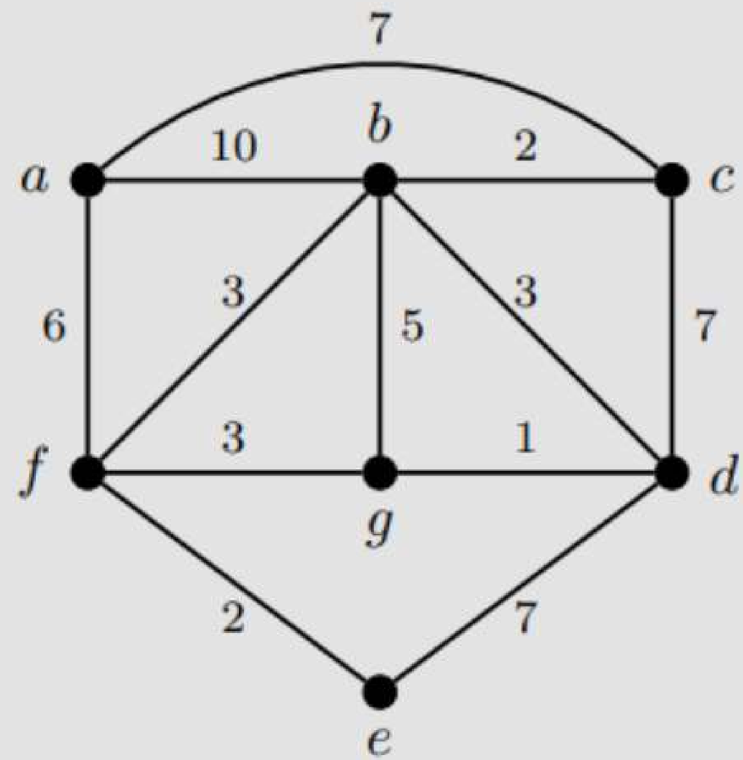
Input: Weighted connected graph $G = (V, E)$.

Steps:

1. Let v be the root. If no root is specified, choose a vertex at random. Highlight it and add it to $T = (V', E')$.
2. Among all edges incident to v, choose the one of minimum weight. Highlight it. Add the edge and its other endpoint to T.
3. Let S be the set of all edges with exactly endpoint from $V(T)$. Choose the edge of minimum weight from S. Add it and its other endpoint to T.
4. Repeat Step (3) until T contains all vertices of G, that is $V(T) = V(G)$.

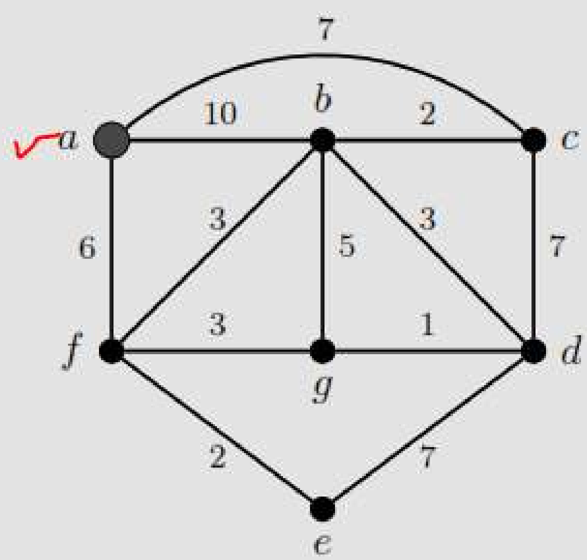
Output: Minimum spanning tree T of G.

Example 3.3 Use Prim's algorithm to find a minimum spanning tree for the graph given in Example 3.2.

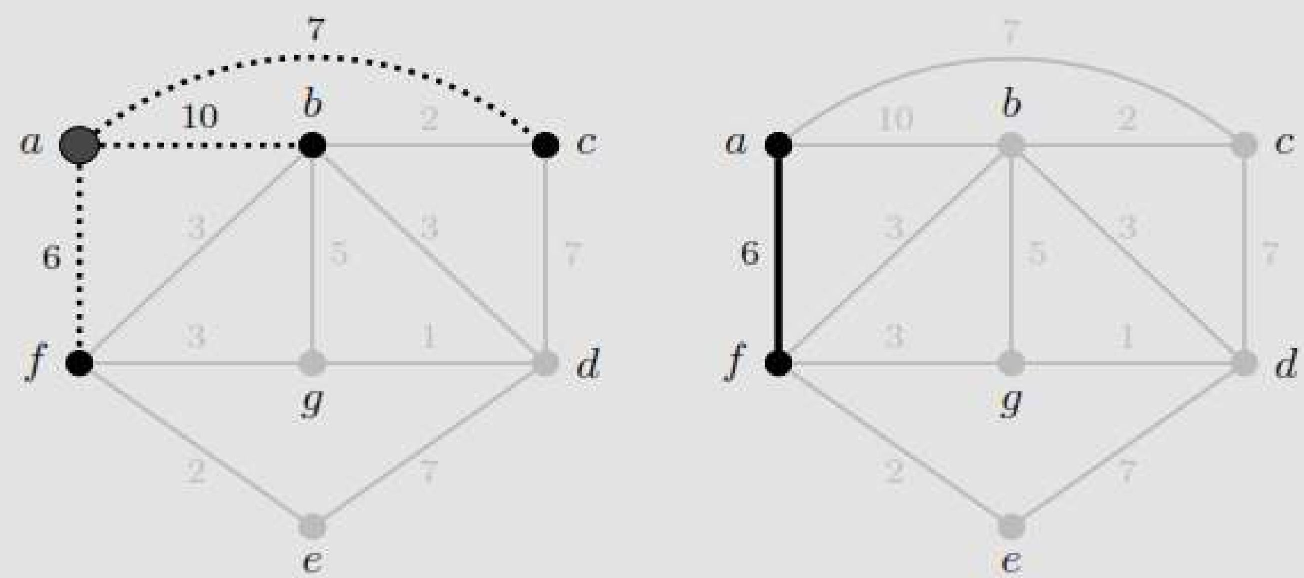


Solution: As before, the newest edges under consideration will be shown as dotted lines and the previously chosen edges will be in black. Unchosen edges will be shown in gray. The vertices in T at each stage will be highlighted as well

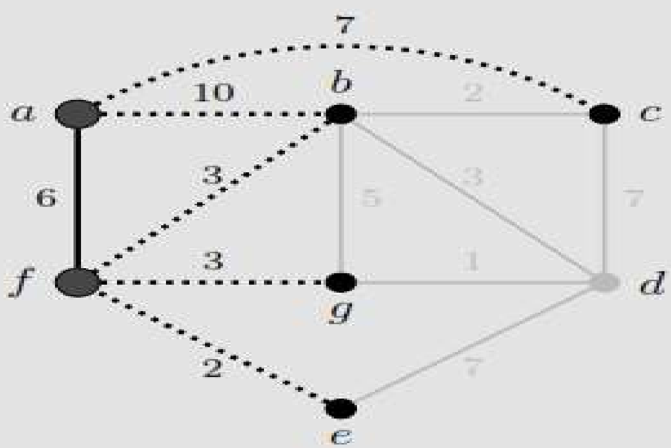
Step 1: Since no root was specified, we choose a as the starting vertex.

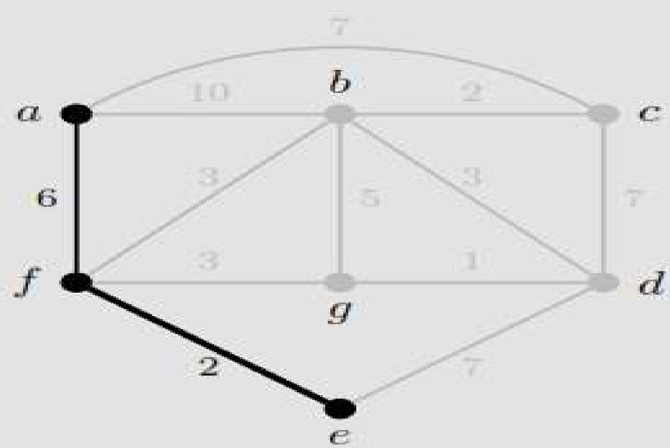


Step 2: We consider the edges incident to a, namely ab, ac, and af. These are shown as dotted lines in the graph on the left. The edge of least weight is af. This is added to the tree, shown in bold on the right.

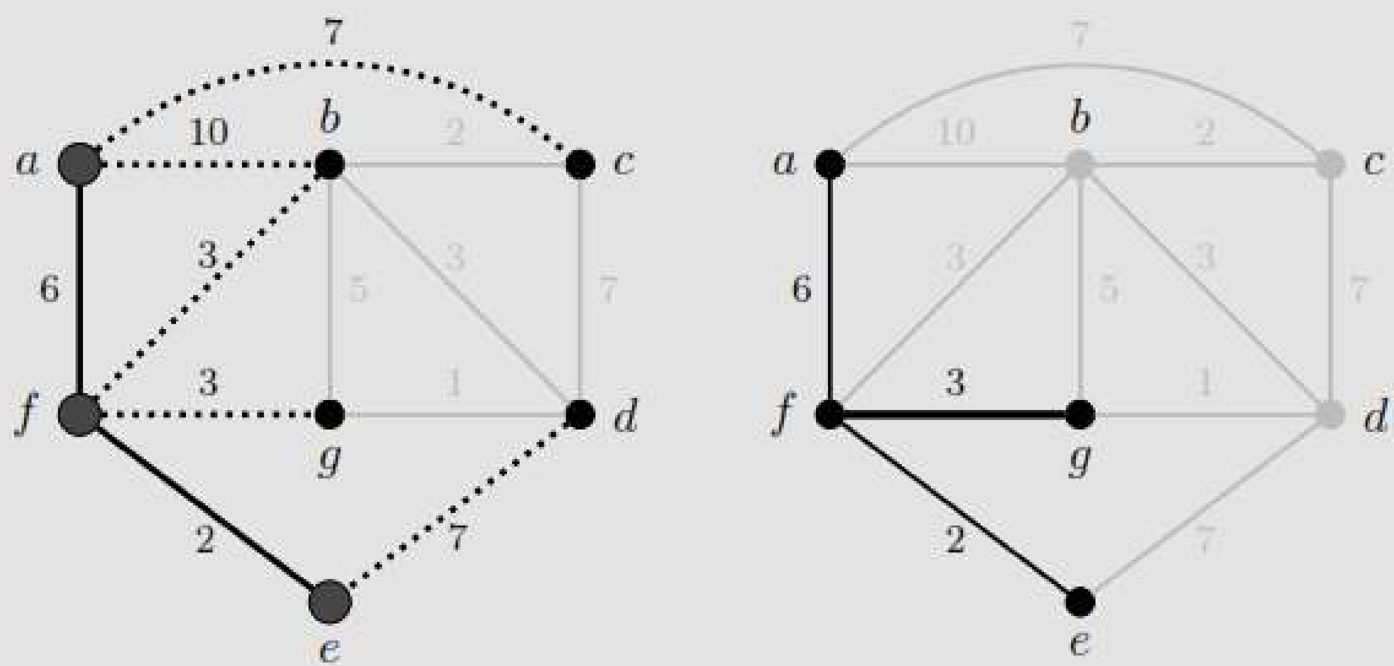


Step 3: The set S consists of edges with one endpoint as a or f, as shown in the graph to the left. The edge of minimum weight from these is ef. This is added to the tree, as shown on the right.

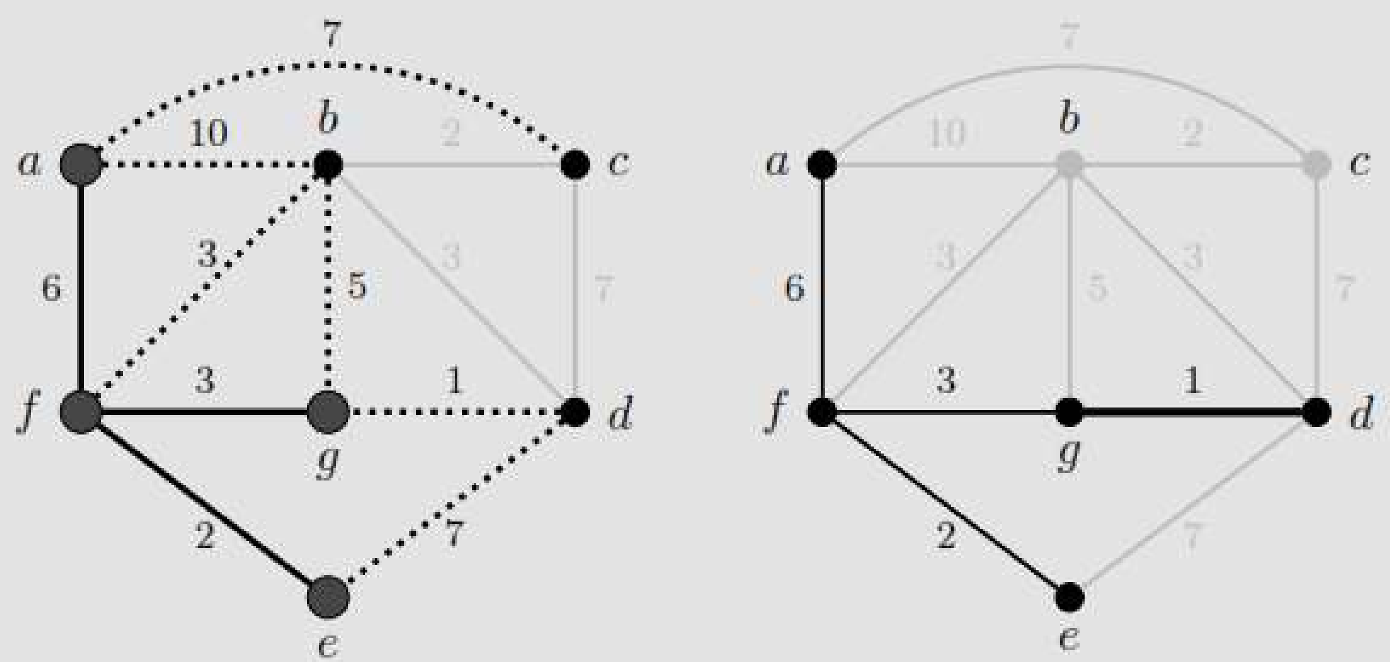




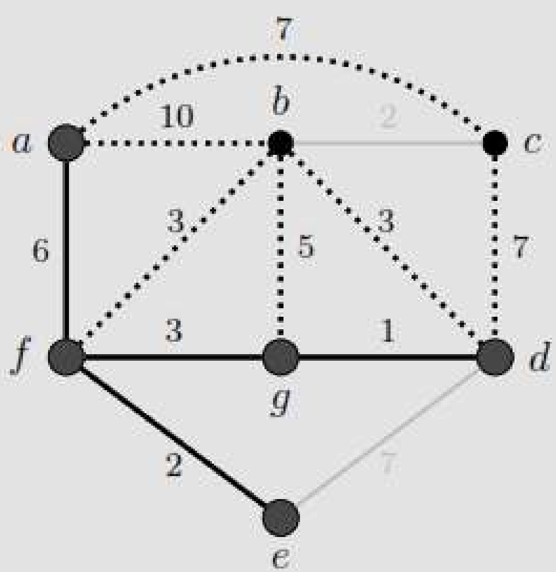
Step 4: The new set S consists of edges with one endpoint as a, e, or f. The next edge added to the tree could either be fg or fb. We choose fg.

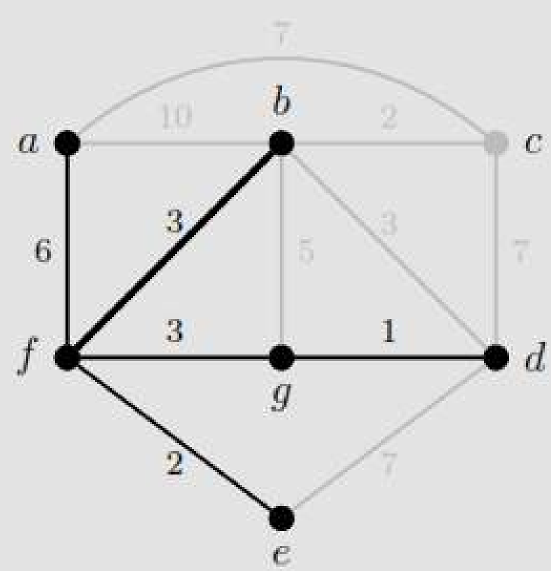


Step 5: We consider the edges where exactly one endpoint is from a, e, f, or g. The next edge to add to the tree is dg.

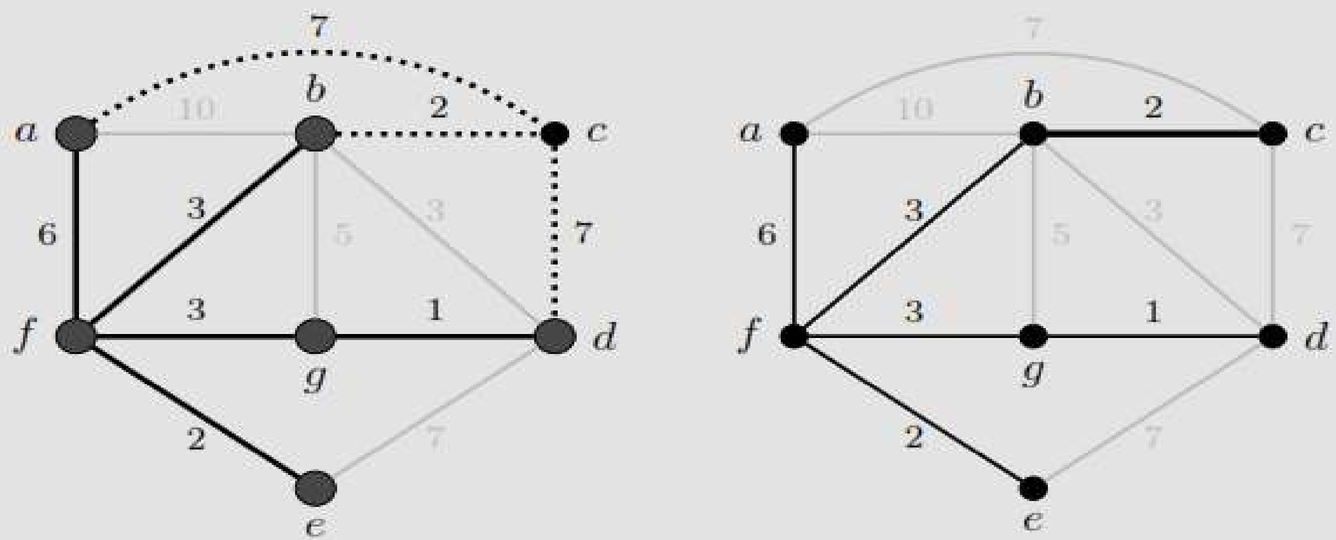


Step 6: The edges to consider must have exactly one endpoint from a, d, e, f, or g. Note that de is no longer available since both endpoints are already part of the tree (and its addition would create a cycle). There are two possible minimum weight edges, bf or bd. We choose bf.

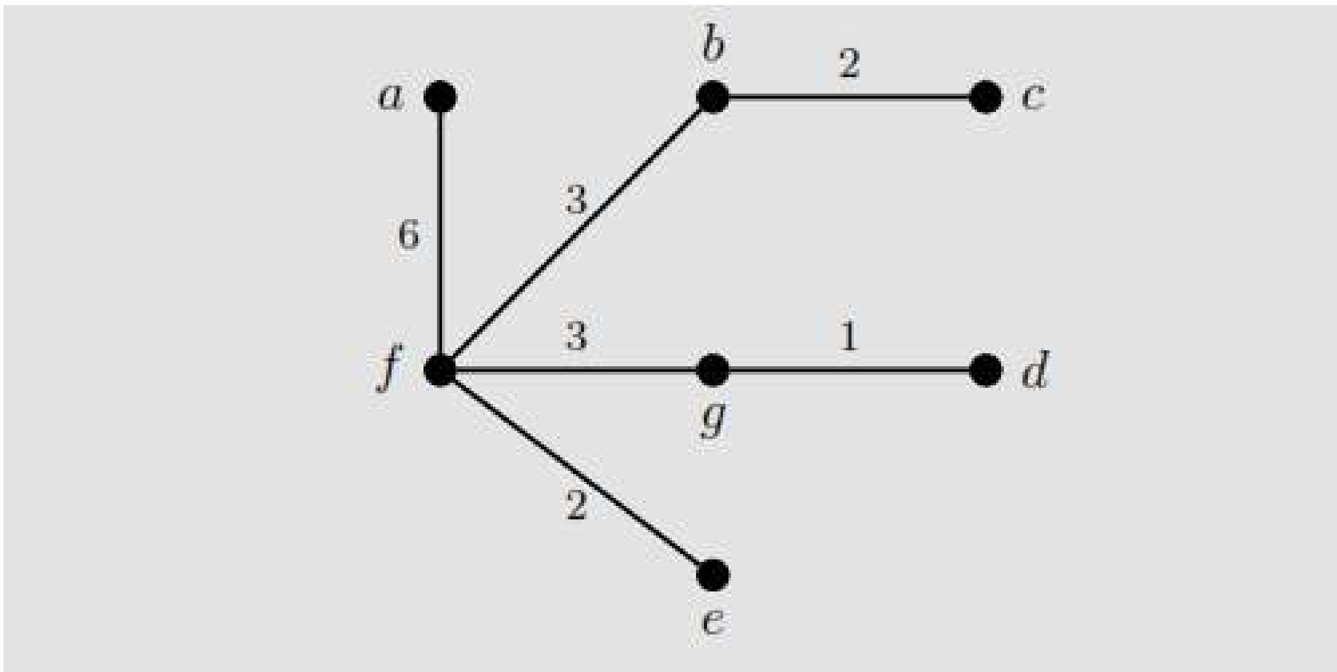




Step 7: The only edges we can consider are those with one endpoint of c since this is the only vertex not part of our tree. The edge of minimum weight is bc.



Output: A minimum spanning tree of total weight 17.


a
b
c
d
e
f
g
2
3
6
3
1
2

Tree Properties

Theorem 3.4 Every tree with at least two vertices has a leaf.

Proof: Suppose for a contradiction that there exists a tree T with at least two vertices that does not contain a leaf. Since T must be connected, we know no vertex has degree 0, and therefore every vertex of T must have degree at least 2. But then by Theorem 2.5 we know T must have a cycle, which contradicts that T is acyclic. Thus T must contain a leaf.

Lemma 3.5 Given a tree T with a leaf v, the graph $T - v$ is still a tree.

Theorem 3.6 A tree with n vertices has $n - 1$ edges for all $n \geq 1$.

Corollary 3.7 The total degree of a tree on n vertices is $2n - 2$.

Proposition 3.8 Let T be a tree. Then for every pair of distinct vertices x and y there exists a unique $x - y$ path.

Proposition 3.9 Every tree is minimally connected, that is the removal of any edge disconnects the graph.

Proposition 3.10 If any edge e is added to a tree T then $T + e$ contains exactly one cycle.

Theorem 3.12 Kruskal's Algorithm produces a minimum spanning tree.

Theorem 3.11 Let T be a graph with n vertices. The following conditions are equivalent:

(a) T is a tree.

(b) T is acyclic and contains $n - 1$ edges.

(c) T is connected and contains $n - 1$ edges.

(d) There is a unique path between every pair of distinct vertices in T.

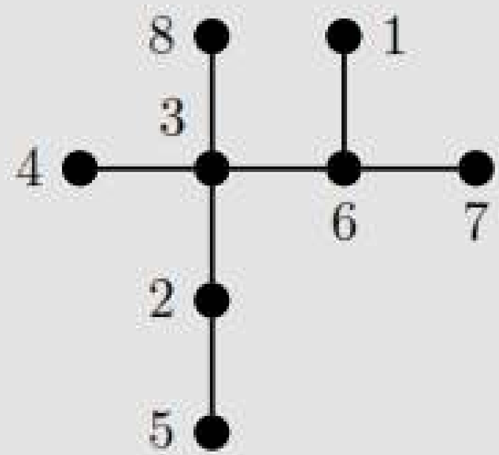
(e) Every edge of T is a bridge.

(f) T is acyclic and for any edge e from T, $T + e$ contains exactly one cycle.

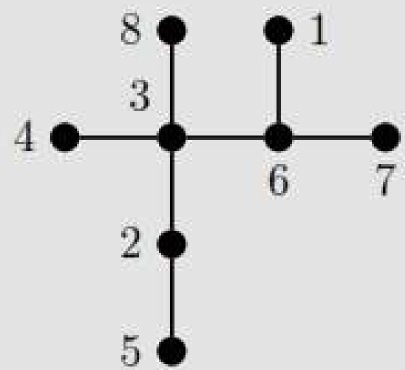
Definition 3.13 Given a tree T on $n > 2$ vertices (labeled $1, 2, \ldots, n$), the ***Prüfer sequence*** of T is a sequence $(s_1, s_2, \ldots, s_{n-2})$ of length $n-2$ defined as follows:

- Let l_1 be the leaf of T with the smallest label.
- Define T_1 to be $T - l_1$.
- For each $i \geq 1$, define $T_{i+1} = T_i - l_{i+1}$, where l_{i+1} is the leaf with the smallest label of T_i.
- Define s_i to be the neighbor of l_i.

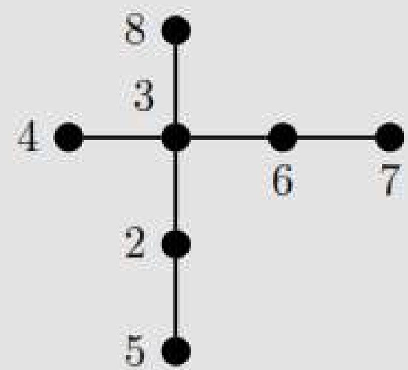
Example 3.4 Find the Prüfer sequence for the tree below.



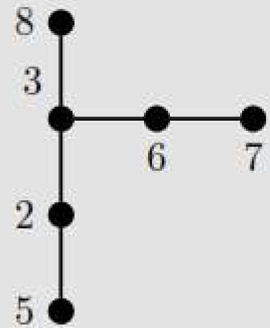
Solution: The pruning of leaves is shown below, with l_i and s_i listed for each step.



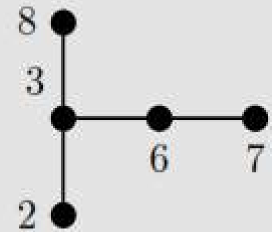
$T: \; l_1 = 1 \quad s_1 = 6$



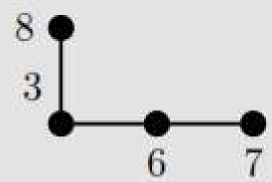
$T_1: \; l_2 = 4 \quad s_2 = 3$



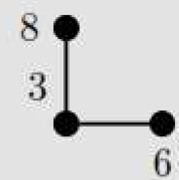
$T_2 : \; l_3 = 5 \quad s_3 = 2$



$T_3 : \; l_4 = 2 \quad s_4 = 3$



$T_4 : \; l_5 = 7 \quad s_5 = 6$



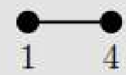
$T_5 : \; l_6 = 6 \quad s_6 = 3$

The Prüfer sequence for the tree T is $(6, 3, 2, 3, 6, 3)$.

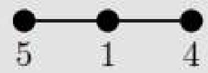
Example 3.5 Find the tree associated to the Prüfer sequence $(1, 5, 5, 3, 2)$.

Example 3.5 Find the tree associated to the Prüfer sequence $(1, 5, 5, 3, 2)$.

Solution: First note that the sequence is of length 5, so the tree must have 7 vertices. At every stage we will consider the possible leaves of a subtree created by the earlier pruning. Initially our set of leaves is $L = \{4, 5, 7\}$, since these do not appear in the sequence and so must be the leaves of the full tree. To begin building the tree, we look for the smallest value not appearing in the sequence, namely 4. This must be adjacent to 1 by entry s_1, as shown on the next page.

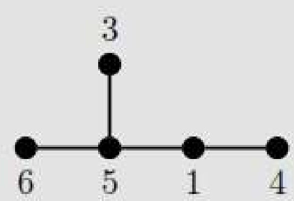


Next, we remove s_1 from the Prüfer sequence and consider the subsequence $(5, 5, 3, 2)$. Now our set of leaves is $L = \{1, 6, 7\}$ since 4 has already been placed as a leaf and 1 is no longer appearing in the sequence. Since 1 is the smallest value in our set L, we know it must be adjacent to $s_2 = 5$, as shown below.

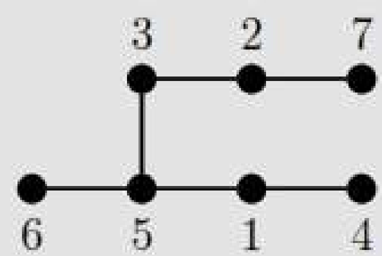


Repeating this process, we see that our subsequence is $(5, 3, 2)$ and $L = \{6, 7\}$. Thus 6 is the leaf adjacent to 5. See the graph below.

Our next subsequence is $(3, 2)$ and $L = \{5, 7\}$, meaning 5 is the smallest value not appearing in the sequence that hasn't already been placed as a "leaf," so there must be and edge from 5 to 3.



Finally we are left with the single entry (2). We know that $L = \{3, 7\}$ and both of these must be adjacent to 2.



To verify we have the correct tree, we can use the process shown in Example 3.4 above to find the Prüfer sequence of our final tree and verify it matches the one given.

Definition 3.15 A ***rooted tree*** is a tree T with a special designated vertex r, called the ***root***. The ***level*** of any vertex in T is defined as the length of its shortest path to r. The ***height*** of a rooted tree is the largest level for any vertex in T.