Chapter 1

Graph Theory

1.1 Graph and Multigraph

A graph G consists of two things:

(i) A set $V = V(G)$ whose elements are called vertices, point or nodes of G.

(ii) A set $E = E(G)$ of unordered pairs of distinct vertices called edges of G.

We denote such a graph by $G(V, E)$.

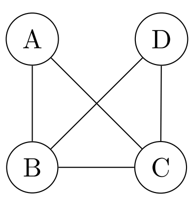


Figure 1.1: G_1

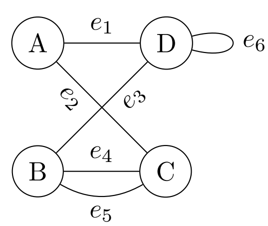


Figure 1.2: G_2

1.2 Multigraphs

Look at the graph 1.2, the edges e_4 and e_5 are called multiple edges since they connect the same endpoints and the edge e_6 is called a loop since its endpoints are the same vertex. Such a diagram is called multigraph.

1.3 Degree

The degree of a vertex v in a graph G written $deg(v)$ is equal to the number of edges in G which contains v, that is which are incident on v. A vertex with degree zero is called isolated. In graph G_2, $deg(D) = 4$, $deg(C) = 3$.

A multigraph is said to be finite if it has a finite number of vertices and a finite number of edges. The finite graph with one vertex and no edges is called the trivial graph.

1.4 Subgraph

Consider a graph $G = G(V, E)$. A graph $H = H(V', E')$ is called a subgraph of G if the vertices and edges of H are contained in the vertices and edges of G.

1.5 Isomorphic Graphs and Homeomorphic Graph

Graphs $G(V, E)$ and $G^*(V^*, E^*)$ are said to be homeomorphic if the can be obtained from the same graph or isomorphic graphs by dividing an edge with additional vertices.

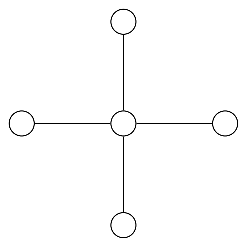


Figure 1.3: Graph A

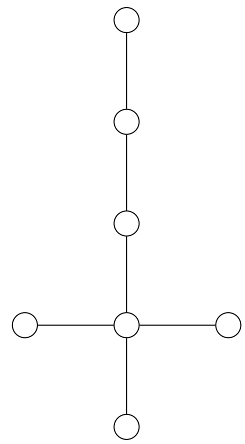


Figure 1.4: Graph B

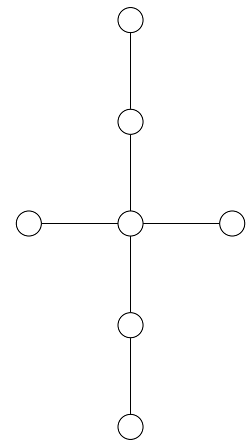


Figure 1.5: Graph C

Graphs 1.4 and 1.5 are homeomorphic since they can be obtained from 1.3 by adding appropriate vertices.

Graphs G and G^* are said to be isomorphic if there exists a one-to-one correspondence $f : V \to V^*$ such that $\{u, v\}$ is an edge of G if and only if $\{f(u), f(v)\}$ is an edge of G^*.

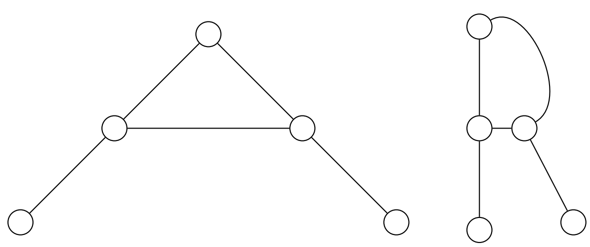


Figure 1.6: Isomorphic Graphs

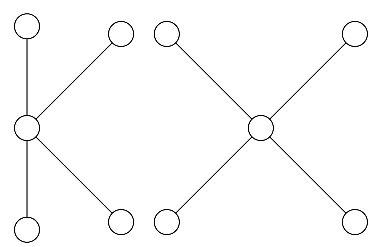


Figure 1.7: Isomorphic Graphs

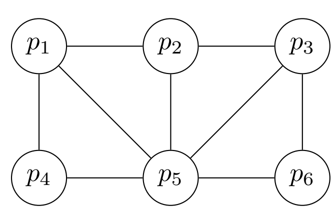
1.6 Paths, Connectivity

A path in a multigraph G consists of vertices and edges of the form,

$v_0,\ e_1,\ v_1,\ e_1,\ v_2,\ e_2, \ldots,\ e_{n-1},\ v_{n-1},\ e_n,\ v_n.$

Where each edge e_i contains the vertices v_{i-1} and v_i. The number n of edges is called the length of the path.

- The path is closed if $v_0 = v_n$.
- A simple path is a path in which all vertices are different.
- A path in which all edges are different will be called a trail.
- A cycle is a closed path in which all vertices are distinct except $v_0 = v_n$.



$\alpha = \{p_4, p_1, p_2, p_5, p_1, p_2, p_3, p_6\}$ is not a trail
$\beta = \{p_4, p_1, p_5, p_2, p_6\}$ is not a path
$\gamma = \{p_4, p_1, p_5, p_2, p_3, p_5, p_6\}$ is not a trail but not simple
$\delta = \{p_4, p_1, p_5, p_3, p_6\}$ is simple but not shortest

- A graph G is connected if there is a path between any two of its vertices.

- The distance between vertices u and v in G written $d(u, v)$ is the length of the shortest path between u and v and the diameter of G, written $diam(g)$ is the maximum distance between any two points in G.

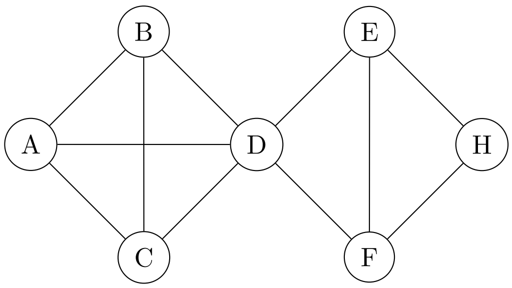


Figure 1.8: G_1

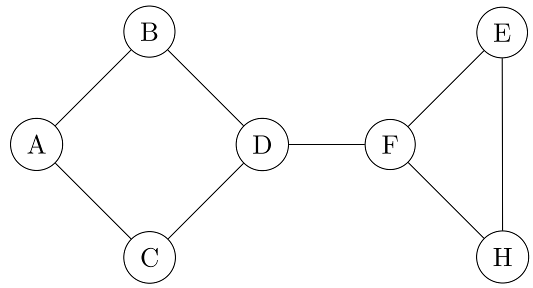


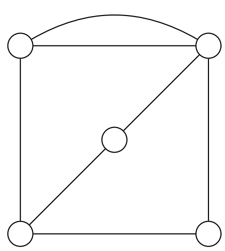
Figure 1.9: G_2

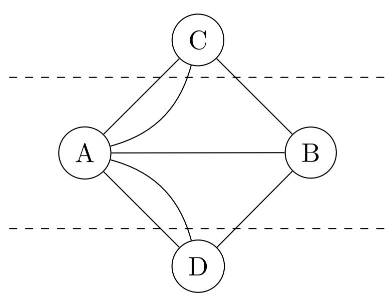
In G_1(figure 1.8), $d(A,\ F) = 2$ and $diam(G_1) = 3$
In G_2(figure 1.9), $d(A,\ F) = 3$ and $diam(G_2) = 4$

- A vertex v in G is called a cutpoint if $G - v$ is disconnected.

- An edge e is called a bridge if $G - e$ is disconnected.
  (D in graph G_1 is cutpoint and $e = \{D, F\}$ is a bridge in G_2.)

- A multigraph is said to be traversable if it can be drawn without any breaks in the curve and without repeating any edges.





A multigraph with more than two odd vertices cannot be traversable. The famous königsberg bridge problem has four odd vertices.

- A graph G is called an Eulerian graph if there exists a closed traversable trail.

- A finite connected graph is Eulerian if and only if each vertex has even degree.

- A Hamiltonian circuit in a graph G named after the nineteenth-century Irish mathematician William Hamilton, is a closed path that visits every vertex in G exactly once.

  A graph G that admits a Hamiltonian circuit is called a Hamiltonian graph.

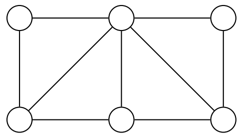


Figure 1.10: Hamiltonian and non-Eulerian

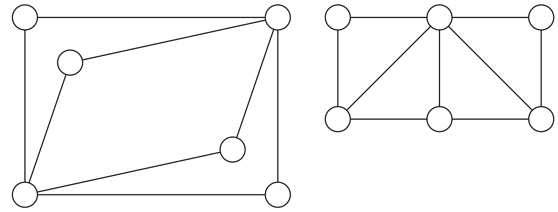


Figure 1.11: Eulerian and non-Hamilton

- A graph G is called a labeled graph if its edges are assigned data of one kind.

  A graph G is called a weighted graph if each e of G is assigned a non-negative number $w(e)$ called the weight or length of v.

- A graph G is said to be complete if every vertex in G is connected to every other vertex in G. The complete graph with n vertices is denoted by k_n.

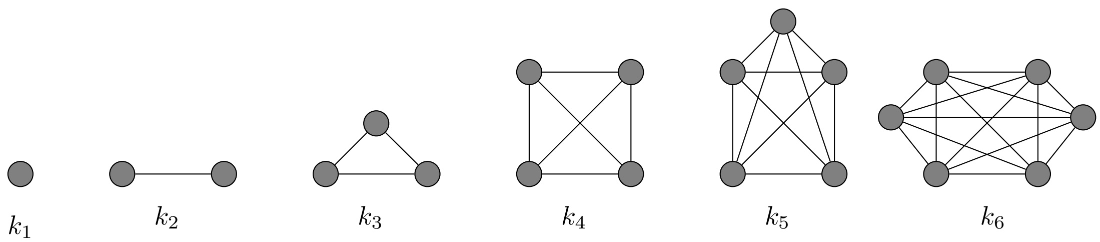


Figure 1.12: Some complete graphs

- A graph G is k-regular if every vertex has degree k.

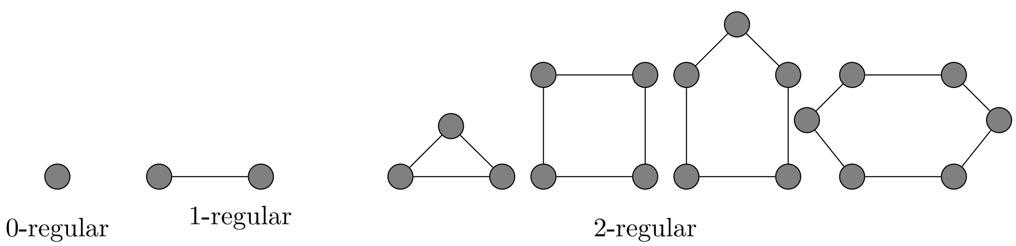
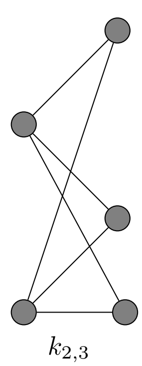


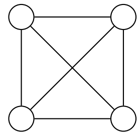
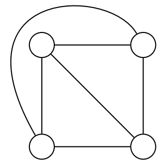
Figure 1.13: Some regular graphs

- A graph G is said to be bipartite if its vertices V can be partitioned into two subsets M and N such that each edge of G connects a vertex of M to a vertex of N.

  By $k_{m,n}$ we mean that each vertex of M is connected to each vertex of N, a complete bipartite graph.



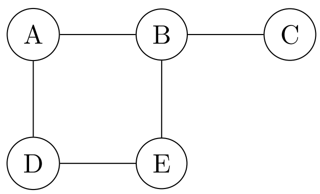
- A graph or multigraph which can be drawn in the plane so that its edges do not cross is said to be planar.

- A particular planar representation of a finite planar multigraph is called a map.
- Let G be a connected planar graph with p vertices and q edges, where $p \geq 3$. Then $q \leq 3p - 6$.
- Suppose G is a graph with m vertices and suppose the vertices have been ordered say, $v_1, v_2, \ldots, v_m$. Then the adjacency matrix $A = [a_{ij}]$ of the graph G is the $m \times m$ matrix defined by

$$a_{ij} = \begin{cases} 1 & \text{if } v_i \text{ is adjacent to } v_j \\ 0 & \text{Otherwise} \end{cases}$$

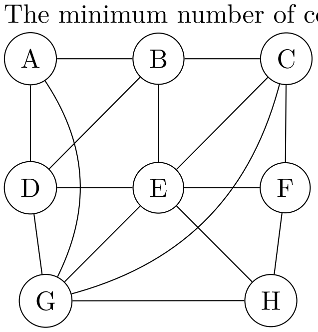
  Adjacency matrix is symmetric.



$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

- A vertex coloring or simply a coloring of G is an assignment of colors to the vertices of G such that adjacent vertices have different colors.

  The minimum number of colors needed to paint G is called the chromatic number of G and is denoted by $\lambda(G)$.



  1. Ordering the vertices of G according to decreasing degrees. Here they are E, G, B, C, D, A, F, H
  2. Assign first color c_1 to first vertex and assign c_1 to each vertex which is not adjacent to first vertex.
  3. Repeat step 2 with second color c_2.
  4. Repeat step 3 and 3 until no vertex left.
  5. Exit.

  First color c_1 to E, A
  Second color c_2 to G, B, F
  Third color c_3 to C, D, H
  Since every vertex is adjacent to every other vertex, k_n requires n colors in any coloring. Thus $\lambda(k_n) = n$.

1.7 Four Color Theorem

Any planar graph is four colorable.

1.7.1 Dual maps and the four color theorem

If the regions of any map M are colored so that adjacent regions have different colors, then no more than four colors are required. Note to self:: TO BE ADDED

1.8 Spanning Tree

A subgraph T of a connected graph G is called a spanning tree of G if T is a tree and T includes all the vertices of G.

Suppose G is a connected weighted graph. Then any spanning tree T of G is assigned a total weight obtained by adding the weights of the edges in T.

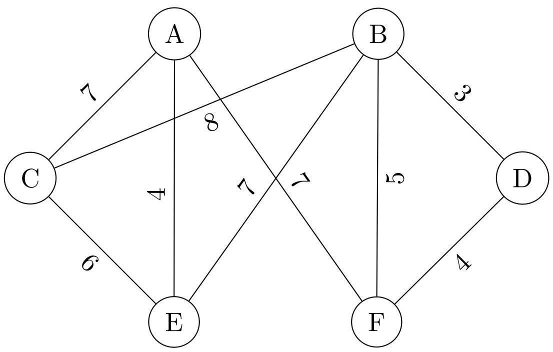
A minimal spanning tree of G is a spanning tree whose total weight is as small as possible.

1.8.1 Kruskal's algorithm for minimal spanning tree

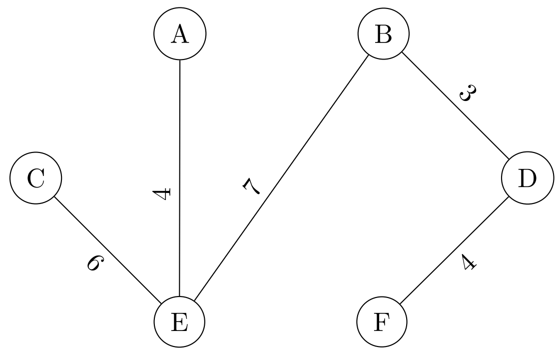
step 1: Arrange the edges of G in order of increasing weights.

step 2: Starting only with the vertices of G and proceeding sequentially, add each edge which does not result in a cycle until $(n-1)$ edges are added.

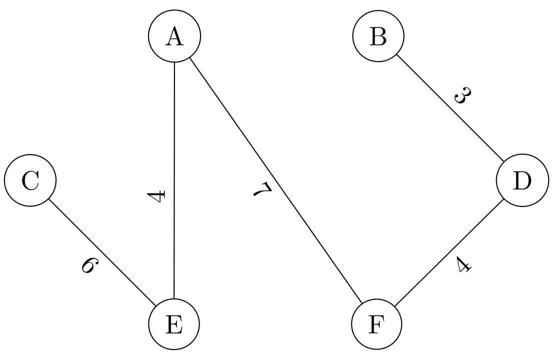
step 3: Exit.



(a)



(b)



(c)

First we order the edges by decreasing weights and then we successively delete edges without disconnecting Q until five edges remain. This yields the following data:

Edges	BC	AF	AC	BE	CE	BF	AE	DF	BD
Weight	8	7	7	7	6	5	4	4	3
Delete	Yes	Yes	Yes	No	No	Yes	No	No	No

Thus the minimal spanning tree of Q which is obtained contains the edges

$$BE,\ CE,\ AE,\ DF,\ BD$$

1.9 Traversing Binary Tree

There are three standard ways of traversing a binary tree T with root R. these three algorithm called pre order, in order and post order.

1.9.1 Pre order

1. Process the root R.
2. Traverse the left subtree of R in pre order.
3. Traverse the right subtree of R in pre order.

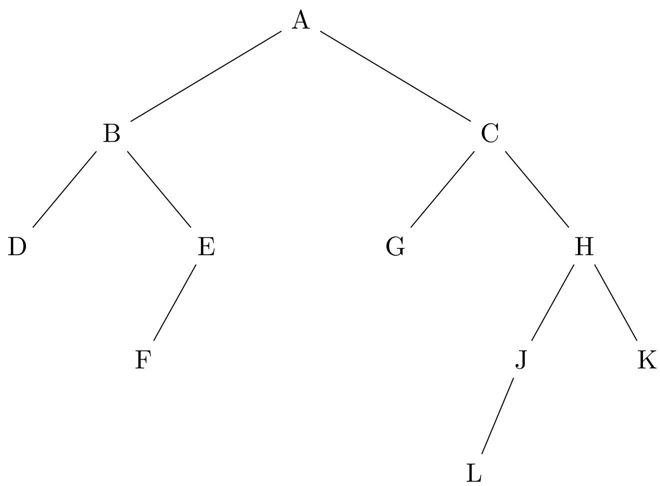
1.9.2 In order

1. Traverse the left subtree of R in in order.
2. Process the root R.
3. Traverse the right subtree of R in in order.

1.9.3 Post order

1. Traverse the left subtree of R in post order.
2. Traverse the right subtree of R in post order.
3. Process the root R.

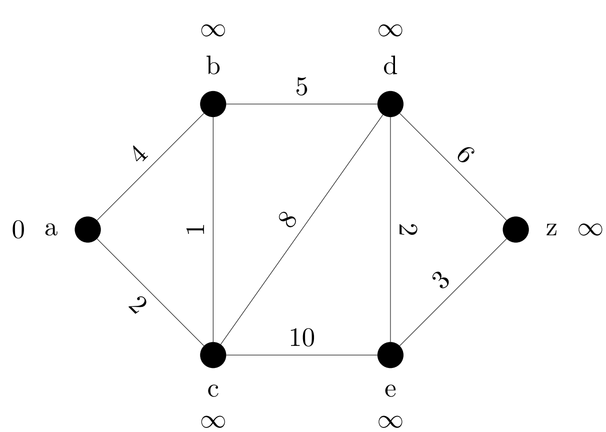
Example. Traverse the tree of the following figure.



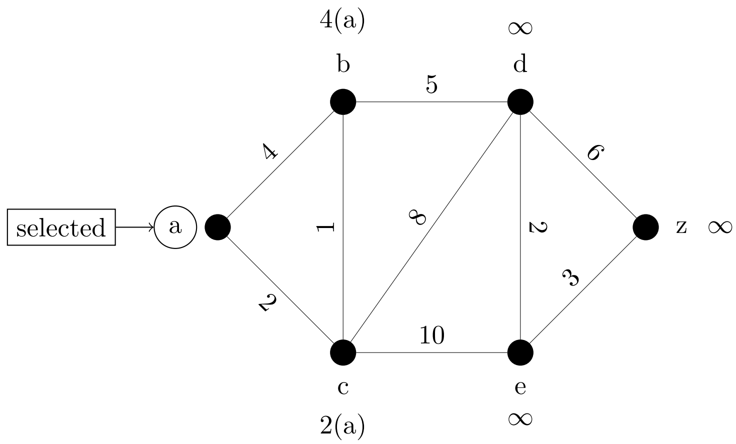
Preorder traversal = A, B, D, E, F, C, G, H, J, L, K
Inorder traversal = D, B, F, E, A, G, C, L, J, H, K
Postorder traversal = D, F, E, B, G, L, J, K, H, C, A

Example. Using Dijkstra's algorithm to find shortest path from a to z.

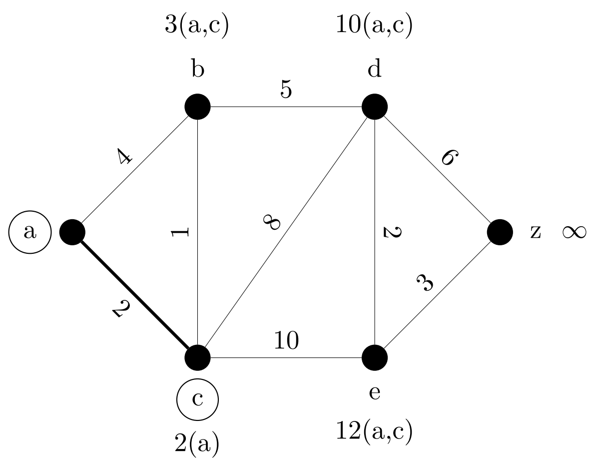
step 1:



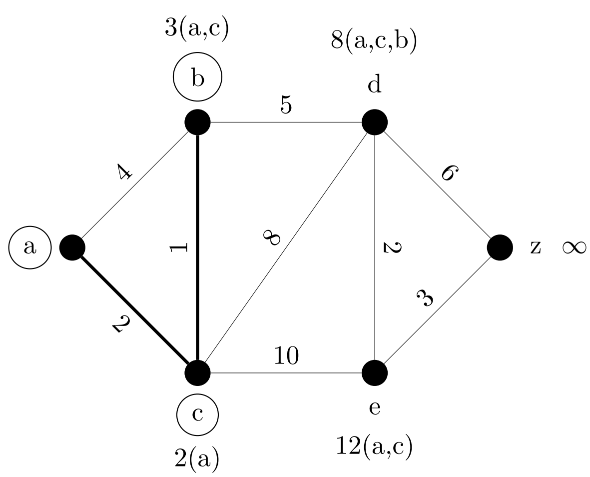
step 2:



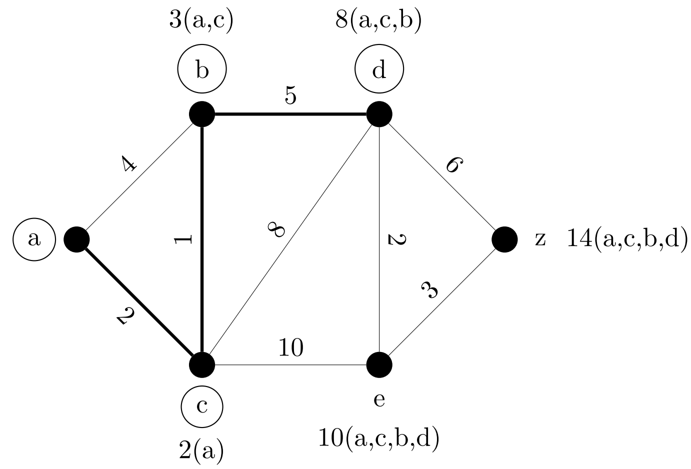
step 3:



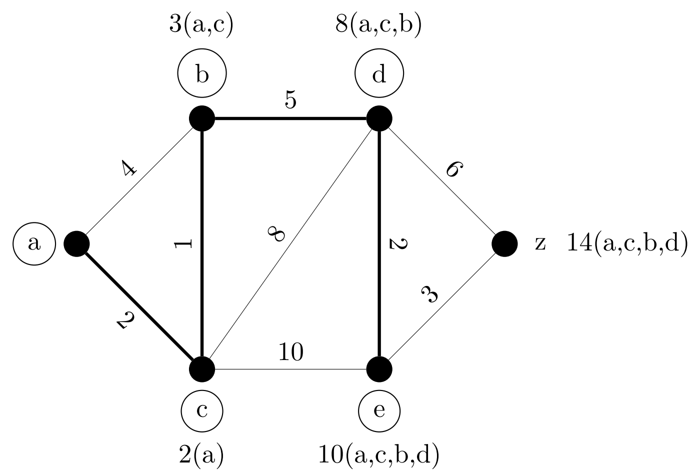
step 4:



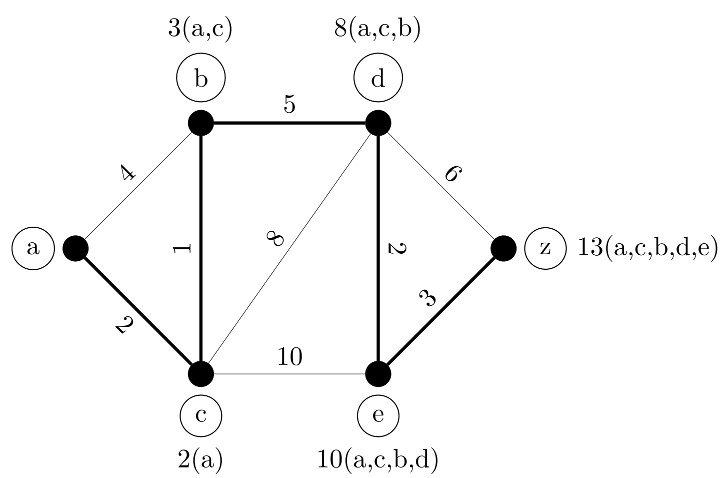
step 5:



step 6:



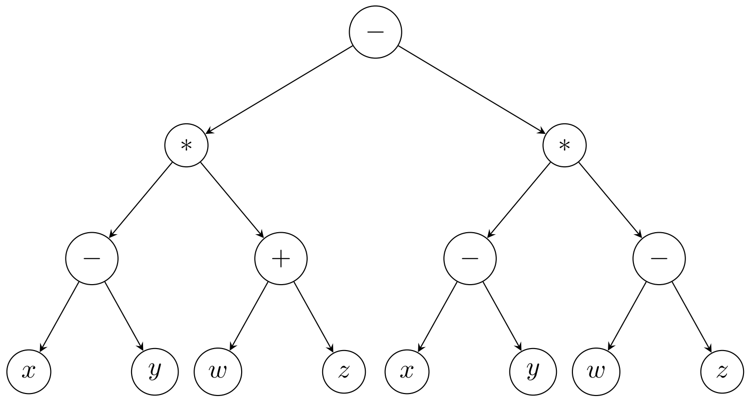
step 7:



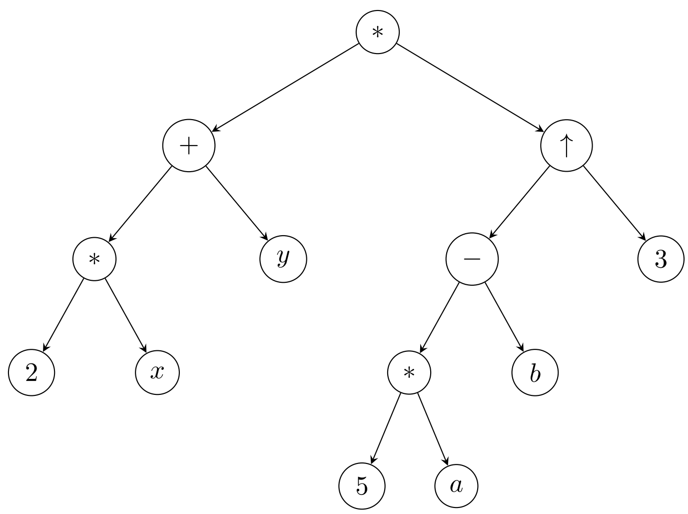
The shortest path: $a \to c \to b \to d \to e \to z$

1.10 Graphical Representation of an Expression

In compiler construction an expression such as $'(x-y)*(w+z)*(x-y)*(w-z)'$ can be represented by the directed acyclic graph[1].



Example. Draw the tree which corresponds to the expression $E=(2x+y)(5a-b)^3$



[1]See tree traversal, infix postfix expression, expression tree.