# Dijkstra's Algorithm

The following tutorial will teach us about Dijkstra's Shortest Path Algorithm. We will understand the working of Dijkstra's Algorithm with a stepwise graphical explanation.

We will cover the following:

* A Brief Overview of the Fundamental Concepts of Graph
* Understand the Use of Dijkstra's Algorithm
* Understand the Working of the Algorithm with a Step-by-Step Example

So, let's get started.

## A Brief Introduction to Graphs

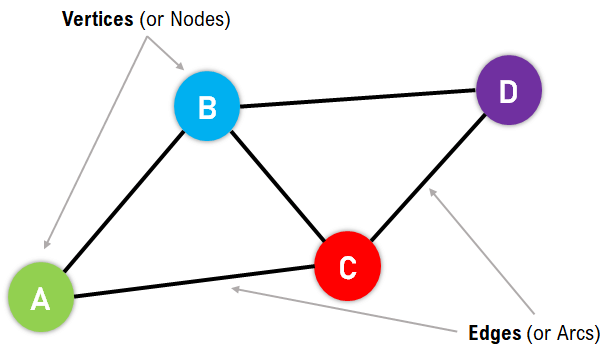
**Graphs** are non-linear data structures representing the "connections" between the elements. These elements are known as the **Vertices**, and the lines or arcs that connect any two vertices in the graph are known as the **Edges**. More formally, a Graph comprises **a set of Vertices (V)** and **a set of Edges (E)**. The Graph is denoted by **G(V, E)**.

ADVERTISEMENT

### Components of a Graph

1. **Vertices:**Vertices are the basic units of the graph used to represent real-life objects, persons, or entities. Sometimes, vertices are also known as Nodes.
2. **Edges:**Edges are drawn or used to connect two vertices of the graph. Sometimes, edges are also known as Arcs.

The following figure shows a graphical representation of a Graph:



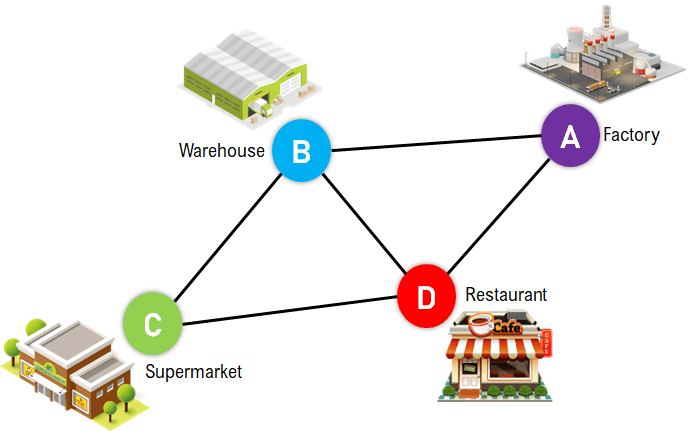
**Figure 1:** Graphical Representation of a Graph

In the above figure, the Vertices/Nodes are denoted with Colored Circles, and the Edges are denoted with the lines connecting the nodes.

### Applications of the Graphs

Graphs are used to solve many real-life problems. Graphs are utilized to represent the networks. These networks may include telephone or circuit networks or paths in a city.

For example, we could use Graphs to design a transportation network model where the vertices display the facilities that send or receive the products, and the edges represent roads or paths connecting them. The following is a pictorial representation of the same:



**Figure 2:** Pictorial Representation of Transportation Network

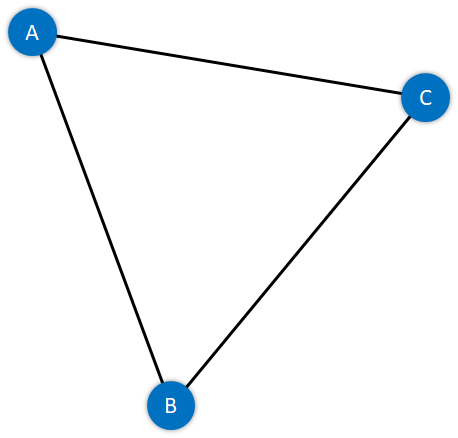
Graphs are also utilised in different Social Media Platforms like LinkedIn, Facebook, Twitter, and more. For example, Platforms like Facebook use Graphs to store the data of their users where every person is indicated with a vertex, and each of them is a structure containing information like Person ID, Name, Gender, Address, etc.

### Types of Graphs

The Graphs can be categorized into two types:

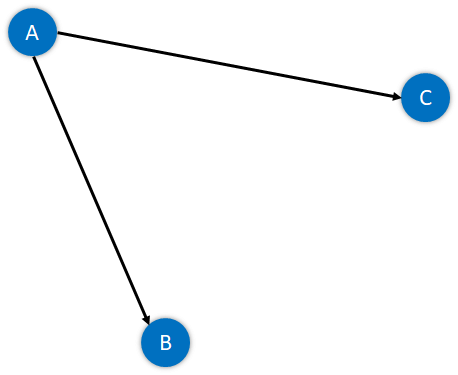
1. Undirected Graph
2. Directed Graph

**Undirected Graph:** A Graph with edges that do not have a direction is termed an Undirected Graph. The edges of this graph imply a two-way relationship in which each edge can be traversed in both directions. The following figure displays a simple undirected graph with 3 nodes and 3 edges.



**Figure 3:** A Simple Undirected Graph

**Directed Graph:** A Graph with edges with direction is termed a Directed Graph. The edges of this graph imply a one-way relationship in which each edge can only be traversed in a single direction. The following figure displays a simple directed graph with 3 nodes and 2 edges.



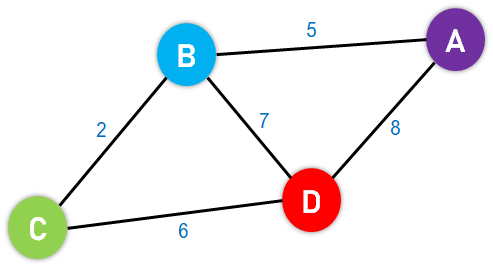
**Figure 4:** A Simple Directed Graph

The absolute length, position, or orientation of the edges in a graph illustration characteristically does not have meaning. In other words, we can visualize the same graph in different ways by rearranging the vertices or distorting the edges if the underlying structure of the graph does not alter.

### What are Weighted Graphs?

A Graph is said to be weighted if each edge is assigned a 'weight'. The weight of an edge can denote distance, time, or anything that models the 'connection' between the pair of vertices it connects.

For instance, we can observe a blue number next to each edge in the following figure of the Weighted Graph. This number is utilized to signify the weight of the corresponding edge.



ADVERTISEMENT

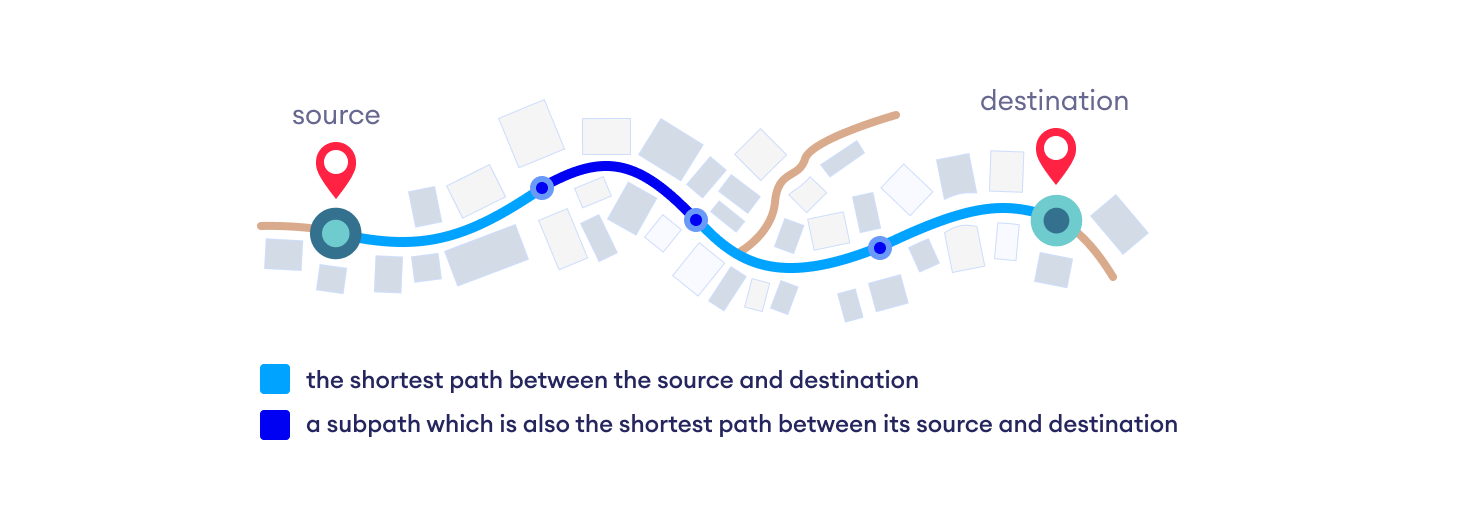
**Figure 5:** An Example of a Weighted Graph

# **Dijkstra's Algorithm**

It differs from the minimum spanning tree because the shortest distance between two vertices might not include all the vertices of the graph.

## **How Dijkstra's Algorithm works**

Dijkstra's Algorithm works on the basis that any subpath B -> D of the shortest path A -> D between vertices A and D is also the shortest path between vertices B and D.

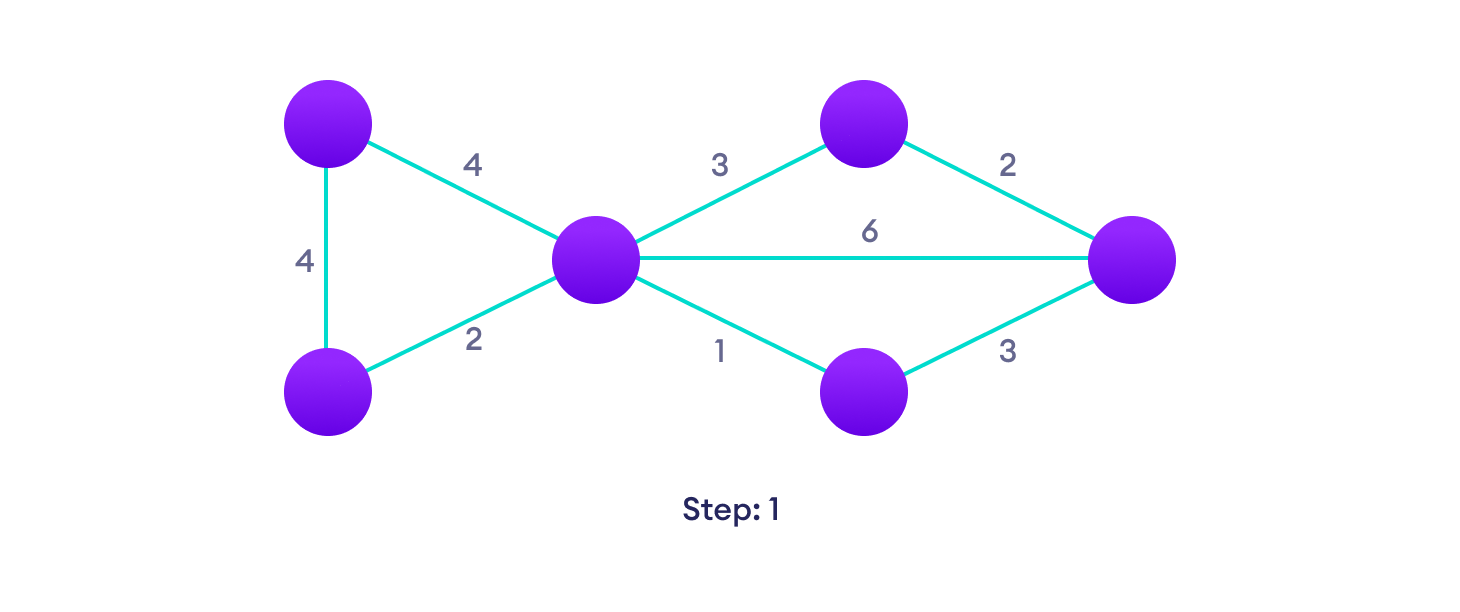
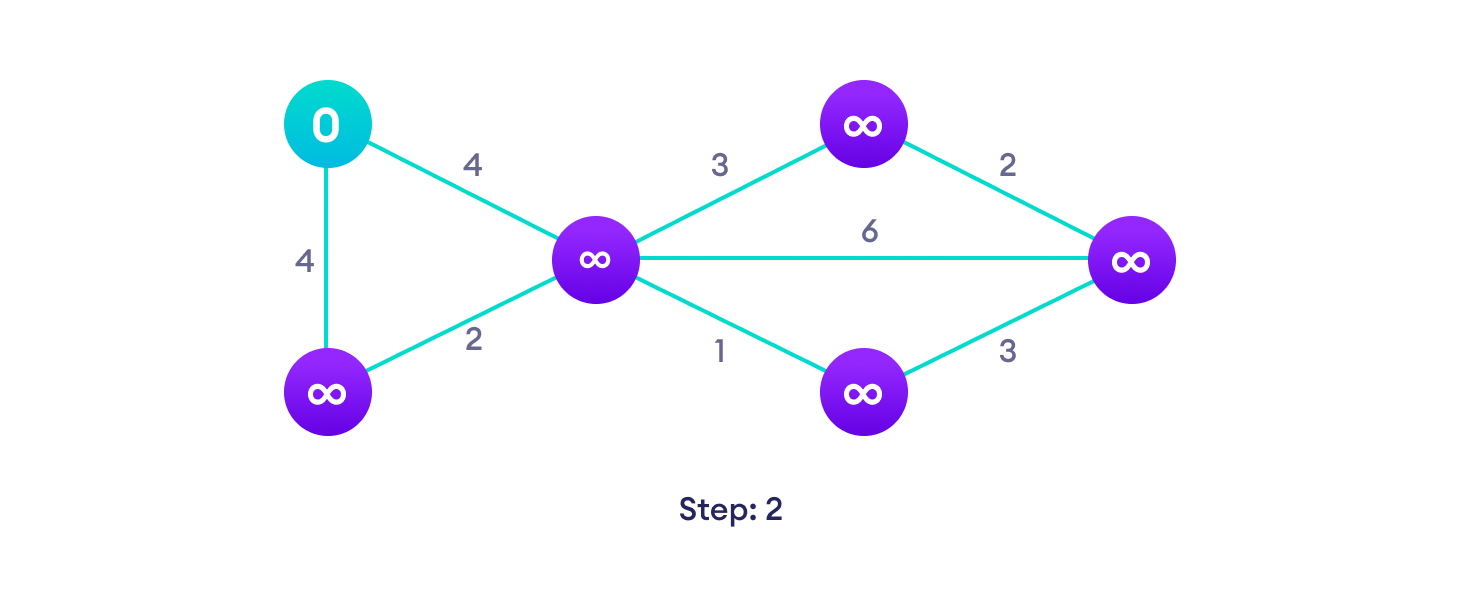
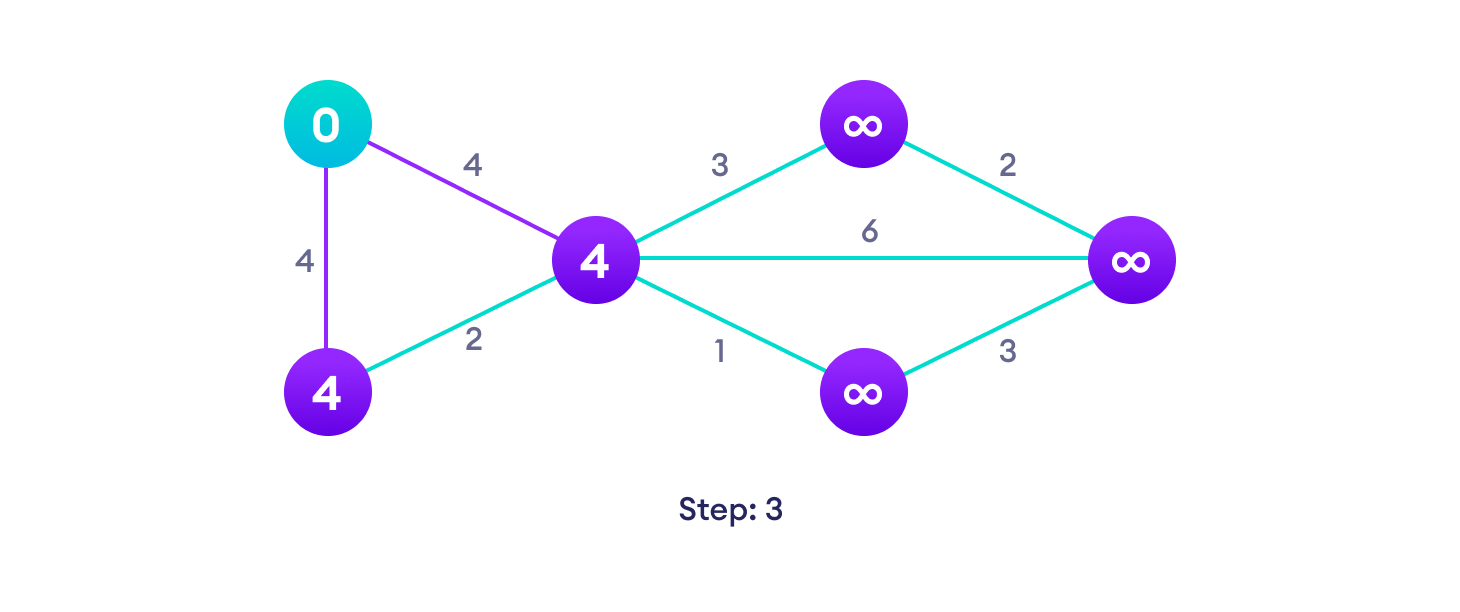
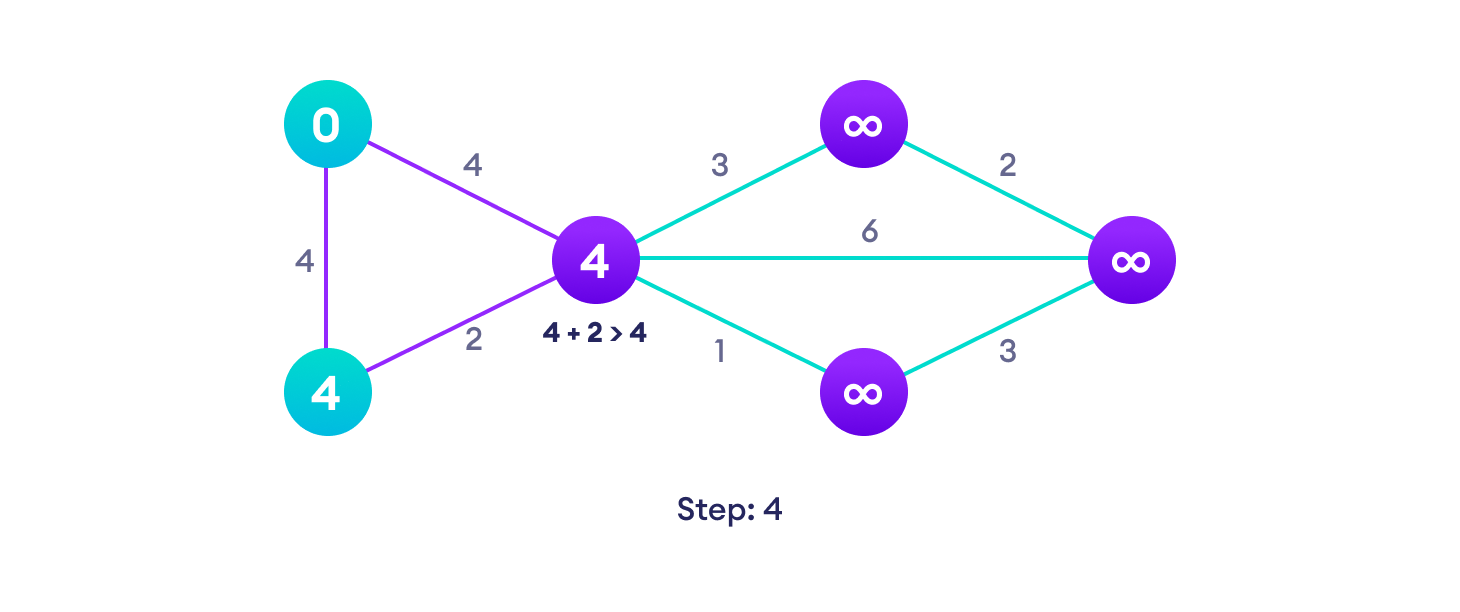
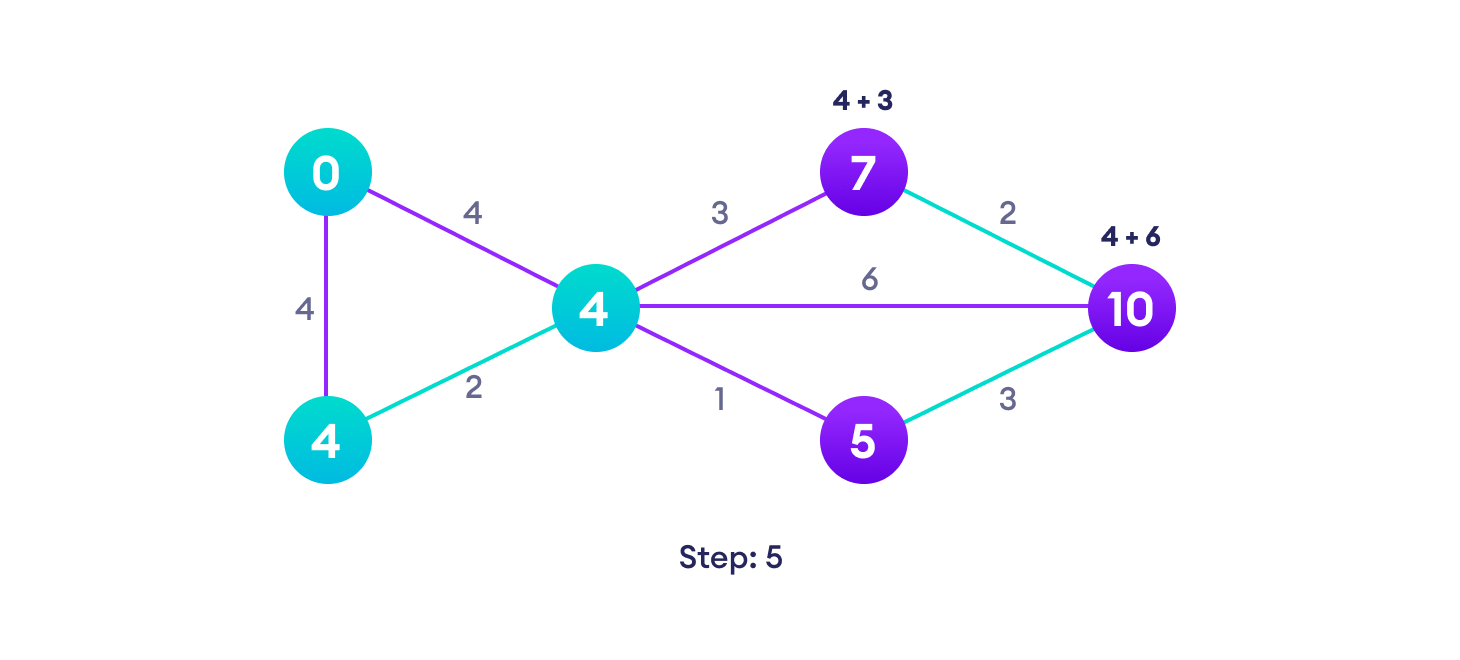
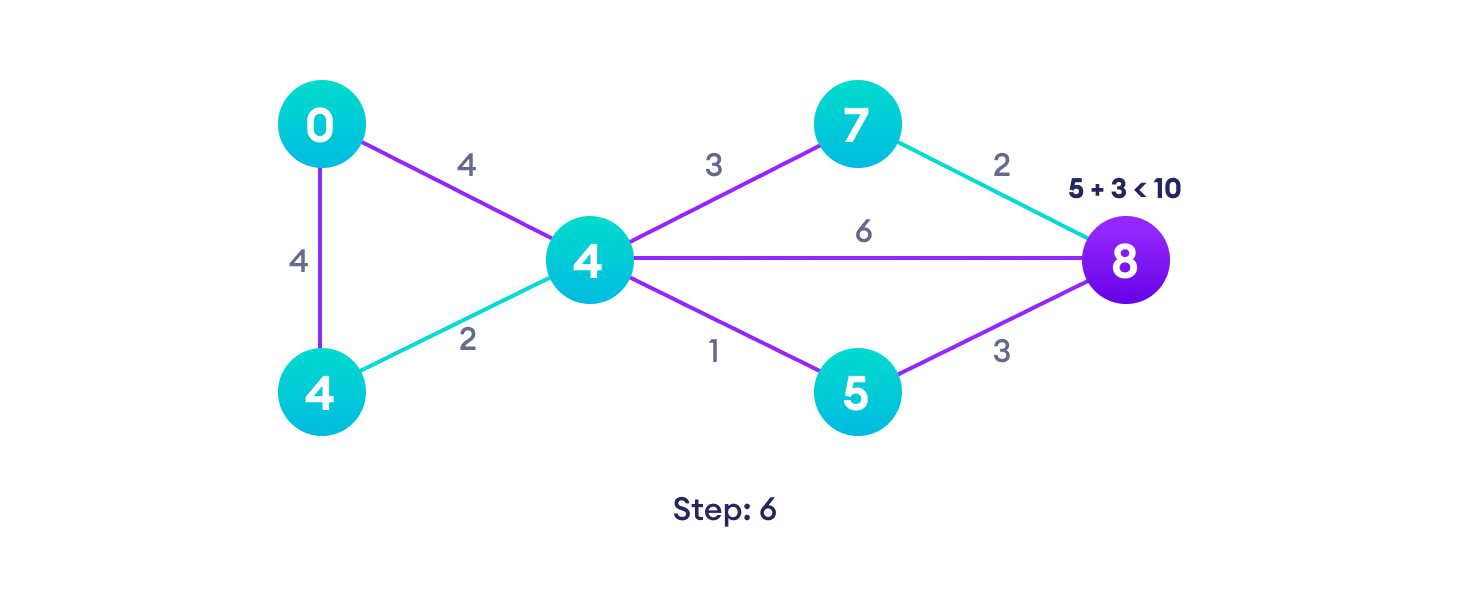
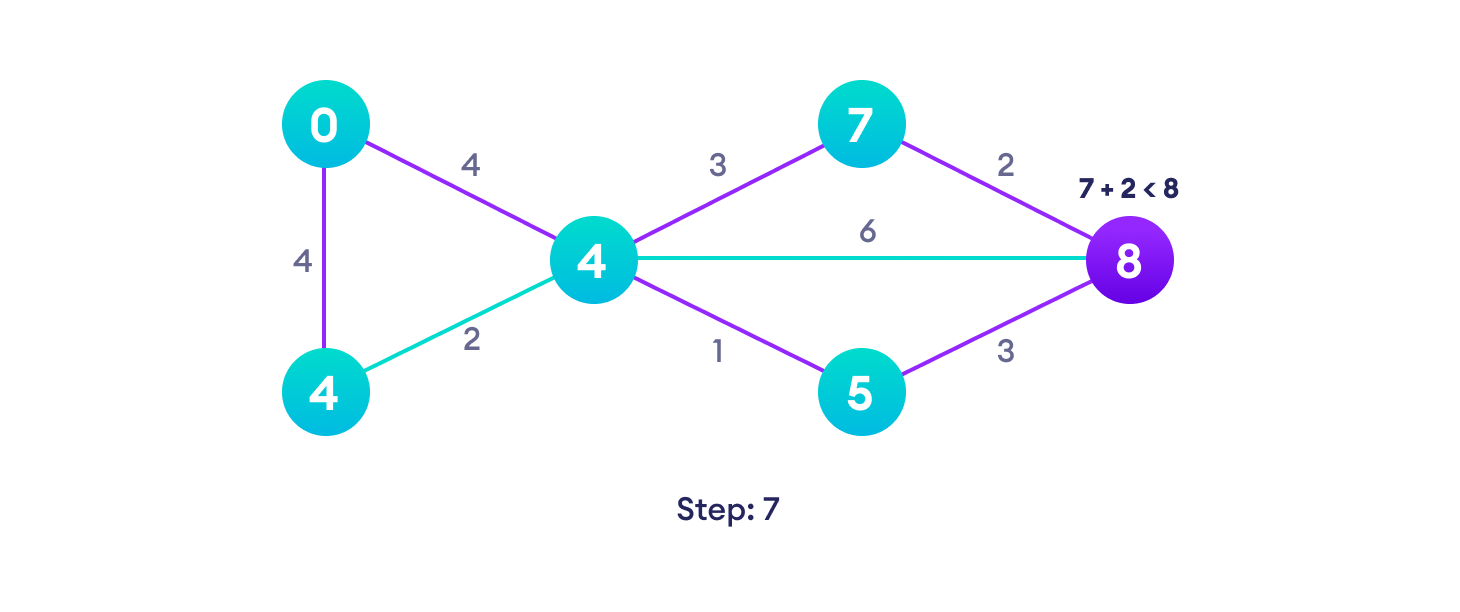
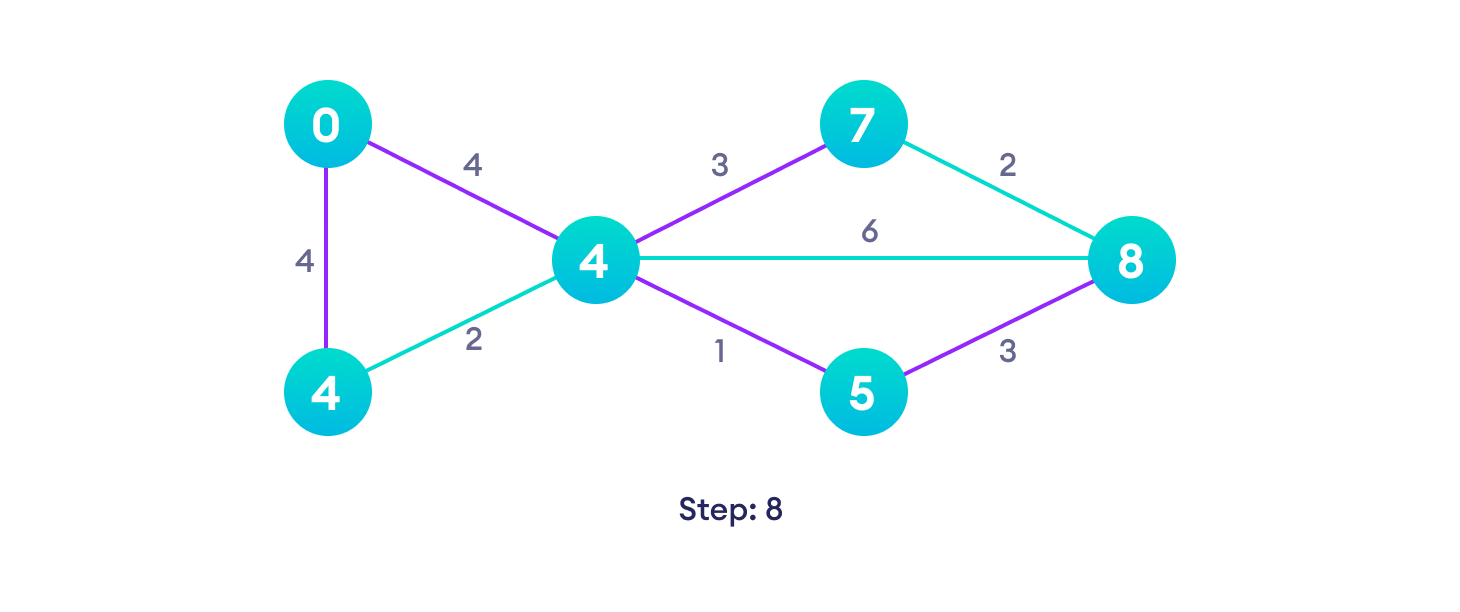
Each subpath is the shortest path

Djikstra used this property in the opposite direction i.e we overestimate the distance of each vertex from the starting vertex. Then we visit each node and its neighbors to find the shortest subpath to those neighbors.

The algorithm uses a greedy approach in the sense that we find the next best solution hoping that the end result is the best solution for the whole problem.

## **Example of Dijkstra's algorithm**

It is easier to start with an example and then think about the algorithm.

Start with a weighted graphChoose a starting vertex and assign infinity path values to all other devicesGo to each vertex and update its path lengthIf the path length of the adjacent vertex is lesser than new path length, don't update itAvoid updating path lengths of already visited verticesAfter each iteration, we pick the unvisited vertex with the least path length. So we choose 5 before 7Notice how the rightmost vertex has its path length updated twiceRepeat until all the vertices have been visited

## **Dijkstra's Algorithm Complexity**

Time Complexity: O(E Log V)

where, E is the number of edges and V is the number of vertices.

Space Complexity: O(V)

## **Dijkstra's Algorithm Applications**

* To find the shortest path
* In social networking applications
* In a telephone network
* To find the locations in the map