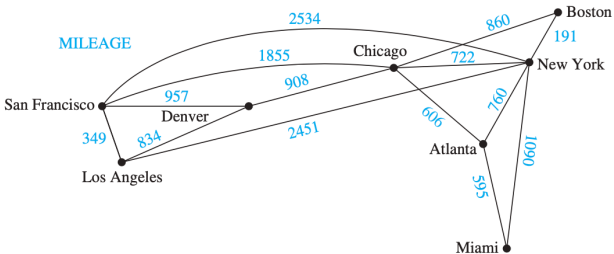
Lecture 5. The Shortest-Path Problem (Section 10.6 and 10.4)

The shortest path problem is the problem of finding a path between two vertices in a graph such that the sum of the weights of its constituent edges is minimized.

Many problems can be modeled using graphs with weights assigned to their edges. As an illustration, consider how an airline system can be modeled.

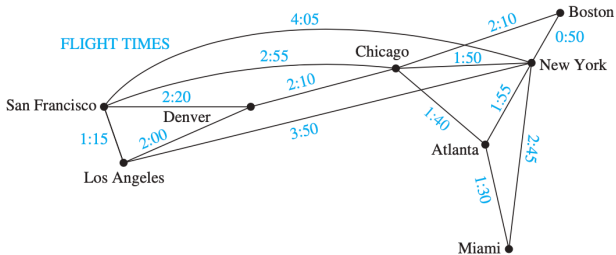


What is a shortest path in *air distance* between San Francisco and Boston?

Graphs that have a number assigned to each edge are called weighted graphs. [1]

[1] Note that the term air distance can be interpreted as the length of a path, which is the sum of all weights on the path. The lenght does not necessarily mean the number of edges in a weighted graph.

Many problems can be modeled using graphs with weights assigned to their edges. As an illustration, consider how an airline system can be modeled.

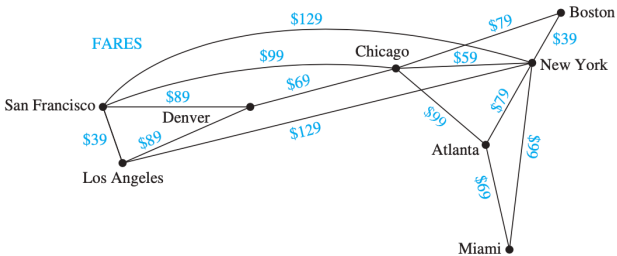


What combinations of flights has the smallest *total flight time* in the air (not including time between flights) between San Francisco and Boston?

Graphs that have a number assigned to each edge are called weighted graphs. [2]

[2] Note that the term total flight time can be interpreted as the length of a path, which is the sum of all weights on the path. The lenght does not necessarily mean the number of edges in a weighted graph.

Many problems can be modeled using graphs with weights assigned to their edges. As an illustration, consider how an airline system can be modeled.



What is the *cheapest fare* between San Francisco and Boston?

Graphs that have a number assigned to each edge are called weighted graphs. [3]

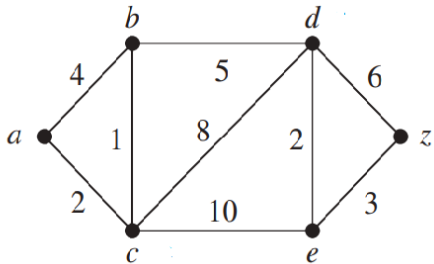
[3] Note that the term cheapest fare can be interpreted as the length of a path, which is the sum of all weights on the path. The lenght does not necessarily mean the number of edges in a weighted graph.

The Shortes-Path Problem

A class of problems that arise in many applications are network flow problems in whcih we wish to find the shortest path between two given vertices in a network.

Dijkstra's Algorithm is a greedy algorithm [4] discovered by the Dutch mathematician Edsger Dijkstra in 1959, which finds the length of a shortest path between two vertices in a connected simple undirected weighted graph.

Example 5.1. What is the length of a shortest path between a and z?



[4] A greedy algorithm is an approach for solving a problem by selecting the best option available at the moment. It doesn't worry whether the current best result will bring the overall optimal result.

ALGORITHM 1 Dijkstra's Algorithm.

procedure *Dijkstra*(G: weighted connected simple graph, with
    all weights positive)
{G has vertices $a = v_0, v_1, \ldots, v_n = z$ and lengths $w(v_i, v_j)$
    where $w(v_i, v_j) = \infty$ if $\{v_i, v_j\}$ is not an edge in G}
for $i := 1$ **to** n
    $L(v_i) := \infty$
$L(a) := 0$
$S := \emptyset$
{the labels are now initialized so that the label of a is 0 and all
    other labels are ∞, and S is the empty set}
while $z \notin S$
    $u :=$ a vertex not in S with $L(u)$ minimal
    $S := S \cup \{u\}$
    for all vertices v not in S
        if $L(u) + w(u, v) < L(v)$ **then** $L(v) := L(u) + w(u, v)$
        {this adds a vertex to S with minimal label and updates the
        labels of vertices not in S}
return $L(z)$ {$L(z) =$ length of a shortest path from a to z}

Visualization of Dijkstra's Algorithm:

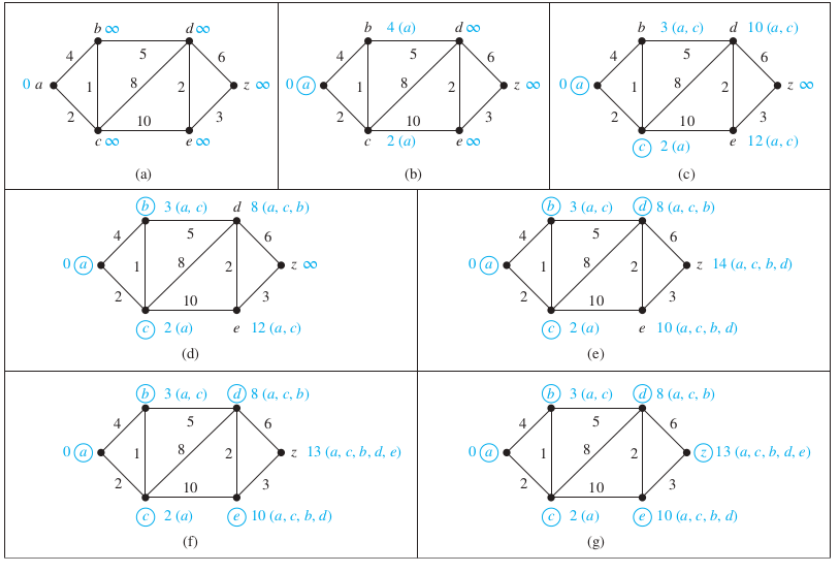
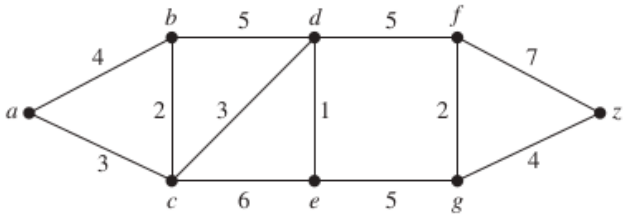


FIGURE 4 **Using Dijkstra's algorithm to find a shortest path from a to z.**

Practice 5.3. Use Dijkstra's algorithm to find a shortest path between a and z in the given weighted graph. Visualize the process as Figure 4 on the previous slide.



Note: It may be instructive to consider how Dijkstra's algorithm can be modified to find the shortest path in a non-simple graph.

Dijkstra's algorithm is fundamental in computer science and graph theory. Understanding and learning to implement it opens doors to more advanced graph algorithms and applications.

It also teaches a valuable problem-solving approach through its **greedy algorithm**, which involves making the optimal choice at each step based on current information.

This skill is transferable to other optimization algorithms. Dijkstra's algorithm may not be the most efficient in all scenarios but it can be a good baseline when solving"shortest distance" problems.

Examples include:

- GPS navigation systems finding the fastest route
- Routing data packets in computer networks
- Delivery services optimizing routes for efficiency
- Social networks (suggesting connections)
- Finance (finding optimal investment paths)
- Project management (finding the most efficient workflow)

Matrix Representation of Graphs: Adjacency Matrix

Matrix Review

$$\begin{bmatrix} 9 & 13 & 5 & 2 \\ 1 & 11 & 7 & 6 \\ 3 & 7 & 4 & 1 \\ 6 & 0 & 7 & 10 \end{bmatrix}$$

$$\mathrm{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

A is an $m \times n$ matrix.
(m rows and n columns.)

If a graph G contains a total of n vertices, we can define an $n \times n$ matrix A by

$$a_{ij} = \begin{cases} w_{ij} & \text{if the length of the edge } [v_i, v_j] = w_{ij} \\ 0 & \text{if there is no edge joining } v_i \text{ and } v_j \end{cases}$$

The resulting matrix $A = [a_{ij}]$ is called the adjacency matrix of the graph G.

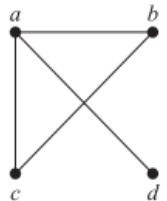
If a graph G contains a total of n vertices, we can define an $n \times n$ matrix A by

$$a_{ij} = \begin{cases} w_{ij} & \text{if the length of the edge } [v_i, v_j] = w_{ij} \\ 0 & \text{if there is no edge joining } v_i \text{ and } v_j \end{cases}$$

The resulting matrix $A = [a_{ij}]$ is called the adjacency matrix of the graph G.

Example 5.4. Use an adjacency matrix to represent the graph.

We order the vertices as a, b, c, d. That is, $v_1 = a, v_2 = b, v_3 = c, v_4 = d$. The matrix representing this graph is



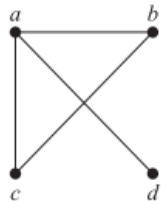
If a graph G contains a total of n vertices, we can define an $n \times n$ matrix A by

$$a_{ij} = \begin{cases} w_{ij} & \text{if the length of the edge } [v_i, v_j] = w_{ij} \\ 0 & \text{if there is no edge joining } v_i \text{ and } v_j \end{cases}$$

The resulting matrix $A = [a_{ij}]$ is called the adjacency matrix of the graph G.

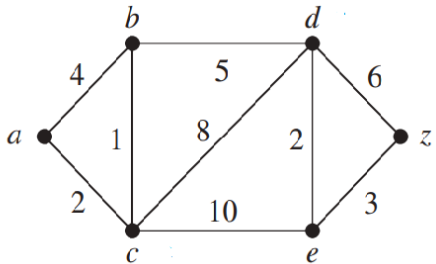
Example 5.4. Use an adjacency matrix to represent the graph.

We order the vertices as a, b, c, d. That is, $v_1 = a, v_2 = b, v_3 = c, v_4 = d$. The matrix representing this graph is

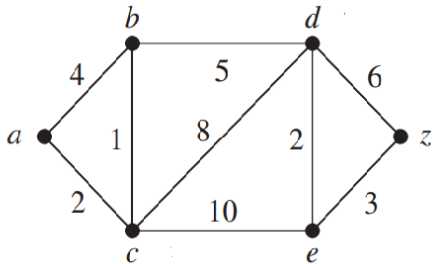


$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Dijkstra's algorithm and Ajacency Matrix



Dijkstra's algorithm and Ajacency Matrix



$$A = \begin{bmatrix} 0 & 4 & 2 & 0 & 0 & 0 \\ 4 & 0 & 1 & 5 & 0 & 0 \\ 2 & 1 & 0 & 8 & 10 & 0 \\ 0 & 5 & 8 & 0 & 2 & 6 \\ 0 & 0 & 10 & 2 & 0 & 3 \\ 0 & 0 & 0 & 6 & 3 & 0 \end{bmatrix}$$

Distance from a

Step	Vertex	a	b	c	d	e	z
1							

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from a

Step	Vertex	a	b	c	d	e	z
1	a						

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a					

Therefore, the shortest-path between a and z is “acbdez of length 13."

Distance from a

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a				

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Therefore, the shortest-path between a and z is “acbdez of length 13."

Distance from a

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a			

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Therefore, the shortest-path between a and z is “acbdez of length 13."

Distance from a

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞		

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Therefore, the shortest-path between a and z is “acbdez of length 13."

Distance from a

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c						

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~					

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c				

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c			

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c		

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c	12c	

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c	12c	∞

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c	12c	∞
		~~0a~~	3c	~~2c~~	10c	12c	∞

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c	12c	∞
		~~0a~~	3c	~~2c~~	10c	12c	∞
3	b						

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c	12c	∞
		~~0a~~	3c	~~2c~~	10c	12c	∞
3	b	~~0a~~					

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c	12c	∞
		~~0a~~	3c	~~2c~~	10c	12c	∞
3	b	~~0a~~	3b				

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c	12c	∞
		~~0a~~	3c	~~2c~~	10c	12c	∞
3	b	~~0a~~	3b	~~2c~~			

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c	12c	∞
		~~0a~~	3c	~~2c~~	10c	12c	∞
3	b	~~0a~~	3b	~~2c~~	8b		

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c	12c	∞
		~~0a~~	3c	~~2c~~	10c	12c	∞
3	b	~~0a~~	3b	~~2c~~	8b	12c	

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c	12c	∞
		~~0a~~	3c	~~2c~~	10c	12c	∞
3	b	~~0a~~	3b	~~2c~~	8b	12c	∞

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c	12c	∞
		~~0a~~	3c	~~2c~~	10c	12c	∞
3	b	~~0a~~	3b	~~2c~~	8b	12c	∞
		~~0a~~	~~3b~~	~~2c~~	8b	12c	∞

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c	12c	∞
		~~0a~~	3c	~~2c~~	10c	12c	∞
3	b	~~0a~~	3b	~~2c~~	8b	12c	∞
		~~0a~~	~~3b~~	~~2c~~	8b	12c	∞
4	d						

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c	12c	∞
		~~0a~~	3c	~~2c~~	10c	12c	∞
3	b	~~0a~~	3b	~~2c~~	8b	12c	∞
		~~0a~~	~~3b~~	~~2c~~	8b	12c	∞
4	d	~~0a~~	~~3b~~	~~2c~~	8d		

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c	12c	∞
		~~0a~~	3c	~~2c~~	10c	12c	∞
3	b	~~0a~~	3b	~~2c~~	8b	12c	∞
		~~0a~~	~~3b~~	~~2c~~	8b	12c	∞
4	d	~~0a~~	~~3b~~	~~2c~~	8d	10d	

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c	12c	∞
		~~0a~~	3c	~~2c~~	10c	12c	∞
3	b	~~0a~~	3b	~~2c~~	8b	12c	∞
		~~0a~~	~~3b~~	~~2c~~	8b	12c	∞
4	d	~~0a~~	~~3b~~	~~2c~~	8d	10d	14d

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c	12c	∞
		~~0a~~	3c	~~2c~~	10c	12c	∞
3	b	~~0a~~	3b	~~2c~~	8b	12c	∞
		~~0a~~	~~3b~~	~~2c~~	8b	12c	∞
4	d	~~0a~~	~~3b~~	~~2c~~	8d	10d	14d
		~~0a~~	~~3b~~	~~2c~~	~~8d~~	10d	14d

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c	12c	∞
		~~0a~~	3c	~~2c~~	10c	12c	∞
3	b	~~0a~~	3b	~~2c~~	8b	12c	∞
		~~0a~~	~~3b~~	~~2c~~	8b	12c	∞
4	d	~~0a~~	~~3b~~	~~2c~~	8d	10d	14d
		~~0a~~	~~3b~~	~~2c~~	~~8d~~	10d	14d
5	e						

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c	12c	∞
		~~0a~~	3c	~~2c~~	10c	12c	∞
3	b	~~0a~~	3b	~~2c~~	8b	12c	∞
		~~0a~~	~~3b~~	~~2c~~	8b	12c	∞
4	d	~~0a~~	~~3b~~	~~2c~~	8d	10d	14d
		~~0a~~	~~3b~~	~~2c~~	~~8d~~	10d	14d
5	e	~~0a~~	~~3b~~	~~2c~~	~~8d~~		

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c	12c	∞
		~~0a~~	3c	~~2c~~	10c	12c	∞
3	b	~~0a~~	3b	~~2c~~	8b	12c	∞
		~~0a~~	~~3b~~	~~2c~~	8b	12c	∞
4	d	~~0a~~	~~3b~~	~~2c~~	8d	10d	14d
		~~0a~~	~~3b~~	~~2c~~	~~8d~~	10d	14d
5	e	~~0a~~	~~3b~~	~~2c~~	~~8d~~	10e	13e

Therefore, the shortest-path between a and z is “acbdez of length 13."

$$\begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix}$$

Distance from *a*

Step	Vertex	a	b	c	d	e	z
1	a	0a	4a	2a	∞	∞	∞
		~~0a~~	4a	2a	∞	∞	∞
2	c	~~0a~~	3c	2c	10c	12c	∞
		~~0a~~	3c	~~2c~~	10c	12c	∞
3	b	~~0a~~	3b	~~2c~~	8b	12c	∞
		~~0a~~	~~3b~~	~~2c~~	8b	12c	∞
4	d	~~0a~~	~~3b~~	~~2c~~	8d	10d	14d
		~~0a~~	~~3b~~	~~2c~~	~~8d~~	10d	14d
5	e	~~0a~~	~~3b~~	~~2c~~	~~8d~~	10e	13e
		~~0a~~	~~3b~~	~~2c~~	~~8d~~	~~10e~~	13e

Therefore, the shortest-path between a and z is “acbdez of length 13."

Algorithm Complexity of Dijkstra's algorithm = $\mathcal{O}(N^2)$

When analyzing an algorithm for solving the shortest-path problem, we are interested in determining how much computational effort, or 'work,' the algorithm requires as a function of the number of vertices N in the graph.

Algorithm Complexity of Dijkstra's algorithm = $\mathcal{O}(N^2)$

When analyzing an algorithm for solving the shortest-path problem, we are interested in determining how much computational effort, or 'work,' the algorithm requires as a function of the number of vertices N in the graph. Specifically, we aim to understand how the required work grows as N increases. How much work does Dijkstra's algorithm require?

Algorithm Complexity of Dijkstra's algorithm = $\mathcal{O}(N^2)$

When analyzing an algorithm for solving the shortest-path problem, we are interested in determining how much computational effort, or 'work,' the algorithm requires as a function of the number of vertices N in the graph. Specifically, we aim to understand how the required work grows as N increases. How much work does Dijkstra's algorithm require?

- The set of all shortest routes is built up in N steps, by finding the closest node in Step 1, the next closest node in Step 2, and so on.

Algorithm Complexity of Dijkstra's algorithm = $\mathcal{O}(N^2)$

When analyzing an algorithm for solving the shortest-path problem, we are interested in determining how much computational effort, or 'work,' the algorithm requires as a function of the number of vertices N in the graph. Specifically, we aim to understand how the required work grows as N increases. How much work does Dijkstra's algorithm require?

- The set of all shortest routes is built up in N steps, by finding the closest node in Step 1, the next closest node in Step 2, and so on.
- Each step requires examining the distance from the node that was "solved" in the previous step to all unsolved nodes which can be reached from this node.

Algorithm Complexity of Dijkstra's algorithm = $\mathcal{O}(N^2)$

When analyzing an algorithm for solving the shortest-path problem, we are interested in determining how much computational effort, or 'work,' the algorithm requires as a function of the number of vertices N in the graph. Specifically, we aim to understand how the required work grows as N increases. How much work does Dijkstra's algorithm require?

- The set of all shortest routes is built up in N steps, by finding the closest node in Step 1, the next closest node in Step 2, and so on.
- Each step requires examining the distance from the node that was "solved" in the previous step to all unsolved nodes which can be reached from this node.
- This requires examining at most $N-1$ edges in each of the N steps, so the work required is $N(N-1) = N^2 - N$.

Algorithm Complexity of Dijkstra's algorithm = $\mathcal{O}(N^2)$

When analyzing an algorithm for solving the shortest-path problem, we are interested in determining how much computational effort, or 'work,' the algorithm requires as a function of the number of vertices N in the graph. Specifically, we aim to understand how the required work grows as N increases. How much work does Dijkstra's algorithm require?

- The set of all shortest routes is built up in N steps, by finding the closest node in Step 1, the next closest node in Step 2, and so on.
- Each step requires examining the distance from the node that was "solved" in the previous step to all unsolved nodes which can be reached from this node.
- This requires examining at most $N-1$ edges in each of the N steps, so the work required is $N(N-1) = N^2 - N$.
- We can be expressed the total work required by the algorithm as a function of N such as $\mathcal{O}(N^2)$. (We read $\mathcal{O}$ as "Big-O".)

Algorithm Complexity of Dijkstra's algorithm = $\mathcal{O}(N^2)$

When analyzing an algorithm for solving the shortest-path problem, we are interested in determining how much computational effort, or 'work,' the algorithm requires as a function of the number of vertices N in the graph. Specifically, we aim to understand how the required work grows as N increases. How much work does Dijkstra's algorithm require?

- The set of all shortest routes is built up in N steps, by finding the closest node in Step 1, the next closest node in Step 2, and so on.
- Each step requires examining the distance from the node that was "solved" in the previous step to all unsolved nodes which can be reached from this node.
- This requires examining at most $N-1$ edges in each of the N steps, so the work required is $N(N-1) = N^2 - N$.
- We can be expressed the total work required by the algorithm as a function of N such as $\mathcal{O}(N^2)$. (We read $\mathcal{O}$ as "Big-O".)

Interpretation: An $\mathcal{O}(N^2)$ algorithm requires work that is proportional to N^2, which means that doubling N increases the work required by a factor of 4. (if $N = 10 \nearrow N = 20$, then $10^2 \nearrow 20^2 = 2^2 10^2 = 4(10^2)$)

If an algorithm requires $\mathcal{O}(N^p)$ work for some p, then it is said to be a polynomial time algorithm. The larger p is, the more rapidly the work grows with N. Here are some examples of polynomial time algorithms:

- Finding the largest of N values. Checking the value of each item and keeping track of the largest value seen taken $\mathcal{O}(N)$ operations.
- Solving a system of N linear equations in N unknowns using the Gaussian elimination algorithm takes $\mathcal{O}(N^3)$ operations.
- If the linear system is upper triangular, then it only takes $\mathcal{O}(N^2)$ operations using "back substitution".
- Dijkstra's algorithm for finding the shortest path between two vertices in a network requires $\mathcal{O}(N^2)$.

There are other problems which are essentially much harder, requiring an amount of work which grows exponentially with the size of N, like $\mathcal{O}(2^n)$ or like $\mathcal{O}(N!)$. The Traveling Salesman Problem (TSP) is in this category.

The Traveling Salesman Problem (TSP) asks for an order in which a salesperson should visit each of the cities on his route exactly once so that he travels the minimum total distance. This is equivalent to asking for a Hamilton circuit with minimum total weight in a complete graph, because each vertex is visited exactly once in the circuit.

The most straightforward way to solve an instance of the TSP is to examine all possible Hamilton circuits and select one of minimum total length. How many circuits do we have to examine to solve the problem if there are n vertices in the graph? Answer = $\frac{(n-1)!}{2} \approx \mathcal{O}(n^n)$.

Trying to solve a TSP in this way when there are only a few dozen vertices is impractical. For example, with 25 verices, a total of $24!/2 \approx 3.1 \times 10^{23}$ different Hamilton circuits would have to be considered.

If it took just one nanosecond (10^{-9}) to examine each Hamilton circuit, a total of approximately ten million years would be required to find a minimum-length Hamilton circuit in this graph by exhaustive search techniques.

While many algorithms have been developed which work very well in practice, there is no algorithm known which is guaranteed to solve all TSP's in less than exponential time.