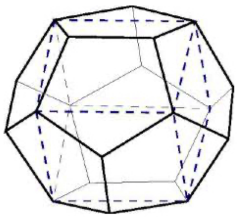
6 Hamiltonian cycle

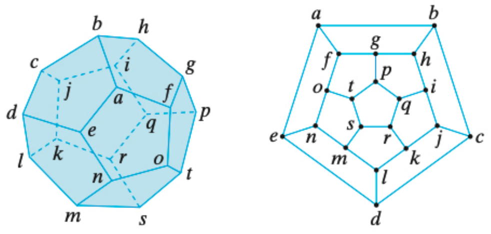
- A cycle in a graph G that contains each vertex in G exactly once, except for the starting and ending vertex that appears twice, ***a Hamiltonian cycle***.

EXAMPLE 6.1

[**Hamilton's puzzle**] The Irish mathematician Sir William Rowan Hamilton devised a puzzle with a regular dodecahedron made of wood. Here is a dodecahedron:



The challenge was to find a route along the edges of the dodecahedron which visited every city exactly once and returned to the start. He labelled each of the vertices with the name of an important city and solved the problem.



The Hamiltonian cycle is shown in following figure.

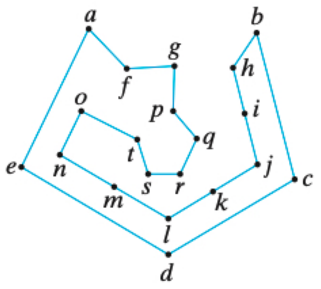


Figure 6.1

EXAMPLE 6.2

Show that the graph of figure 6.2 does not contain a Hamiltonian cycle.

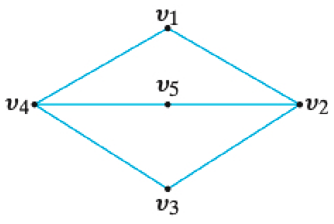


Figure 6.2

EXAMPLE 6.3 Show that the graph in following figure does not contain a Hamiltonian cycle.

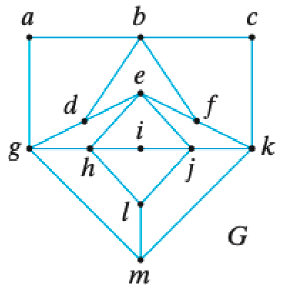


Figure 6.3

- There are no known simple necessary and sufficient criteria for the existence of Hamilton circuits.
- However, many theorems are known that give sufficient conditions for the existence of Hamilton circuits.

Theorem 6.1 – DIRAC'S THEOREM. If G is a simple graph with n vertices with $n \geq 3$ such that the degree of every vertex in G is at least $n/2$, then G has a Hamilton circuit.

Theorem 6.2 – ORE'S THEOREM. If G is a simple graph with n vertices with $n \geq 3$ such that $deg(u) + deg(v) \geq n$ for every pair of nonadjacent vertices u and v in G, then G has a Hamilton circuit.

EXAMPLE 6.4 Verify Ore's theorem by testing to see if the following graph is Hamiltonian.

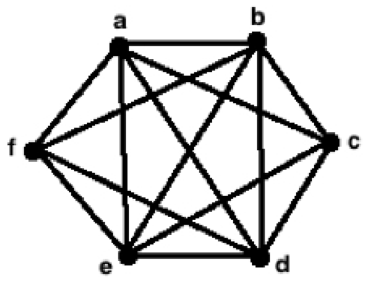


Figure 6.4

Solution

- The graph is simple and $n = 6 \geq 3$.
- There is only one pair of non-adjacent vertices, c and f.
- Then, $deg(c) + deg(f) = 4 + 4 \geq 6$.
- Thus, the graph has a Hamilton circuit and therefore the graph is Hamiltonian.

EXAMPLE 6.5 Let G be a simple graph with n vertices and m edges where m is at least 3. If $m \geq \frac{1}{2}(n-1)(n-2)+2$, prove that G is Hamiltonian. Is the converse true ?

Solution

- Let u and v be any two non adjacent vertices and n_1 and n_2 be their degrees respectively.
- When we elimnate the vertices u, v from graph G, we get a subgraph with $n-2$ vertices.
- This subgraph is a simple graph and if it has q edges then $q \leq \frac{1}{2}(n-2)(n-3)$.[a]
- Since u and v are non adjacent, $m = q + n_1 + n_2$.
- That implies,

$$n_1 + n_2 = m - q \geq \frac{1}{2}(n-1)(n-2) + 2 - \frac{1}{2}(n-2)(n-3 = n$$

- If u and v are two any non adjacent vertices , $deg(u) + deg(v) \geq n$.
- Thus, by the Ore's theorem the graph has a Hamilton circuit and therefore the graph is Hamiltonian.
- The converse of the result just proved is not always true. Because, a 2-regular graph with 5-vertices (see Figure below) is Hamiltonian but the inequality does not hold.

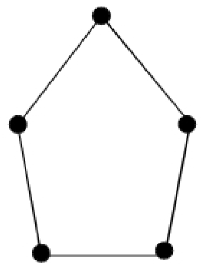


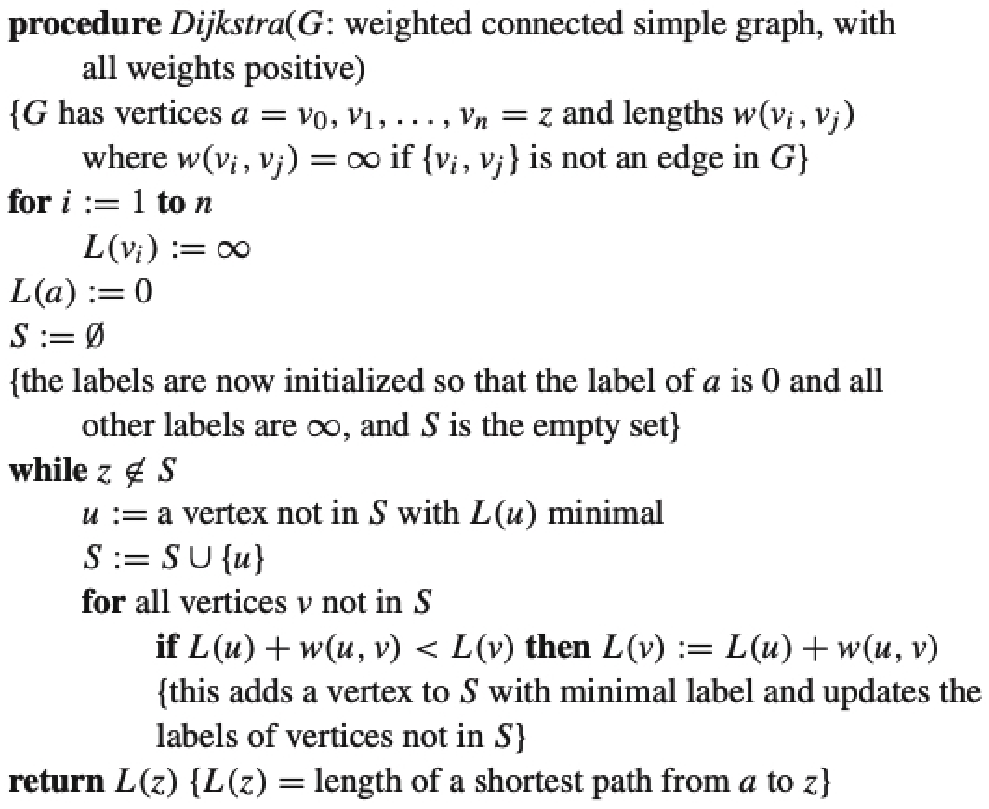
Figure 6.5

[a]The maximum number of edges in a simple graph with n vertices is $\frac{n(n-1)}{2}$.

7 A Shortest-Path Algorithm

- Recall that a weighted graph is a graph in which values are assigned to the edges and that the length of a path in a weighted graph is the sum of the weights of the edges in the path.
- In weighted graphs, we are interesting to find a shortest path (i.e., a path having minimum length) between two given vertices.
- The algorithm to find the shortest path in a weighted graph discovered by the Dutch mathematician Edsger Dijkstra is given below.
- Some Characteristics of Dijkstra's algorithm are
  - Basically, the Dijkstra's algorithm begins from the vertex to be selected, the source vertex, and it examines the entire graph to determine the shortest path among that vertex and all the other vertices in the graph.
  - The algorithm maintains the track of the currently recognized shortest distance from each vertex to the source code and updates these values if it identifies another shortest path.
  - Once the algorithm has determined the shortest path amid the source code to another vertex, the vertex is marked as ”visited” and can be added to the path.
  - This process is being continued till all the vertices in the graph have been added to the path, as this way, a path gets created that connects the source vertex to all the other vertices following the plausible shortest path to reach each vertex.

Algorithm 7.1 – 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 $\infty$, 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 $z$}
```

Theorem 7.1. Dijkstra's algorithm finds the length of a shortest path between two vertices in a connected simple undirected weighted graph.

Steps of Dijkstra's algorithm

Step 1 The very first step is to mark all vertices as unvisited,

Step 2 Mark the picked starting vertex with a current distance of 0 and the rest vertices with infinity,

Step 3 Now, fix the starting vertex as the current vertex,

Step 4 For the current vertex, analyse all of its unvisited neighbours and measure their distances by adding the current distance of the current vertex to the weight of the edge that connects the neighbour vertex and current vertex,

Step 5 Compare the recently measured distance with the current distance assigned to the neighbouring vertex and make it as the new current distance of the neighbouring vertex,

Step 6 After that, consider all of the unvisited neighbours of the current vertex, mark the current vertex as visited, If the destination vertex has been marked visited then stop, an algorithm has ended, and

Step 7 Else, choose the unvisited vertex that is marked with the least distance, fix it as the new current vertex, and repeat the process again from step 4.

EXAMPLE 7.1 Use Dijkstra's algorithm to find the length of a shortest path between the vertices a and z in the weighted graph displayed in Figure below.

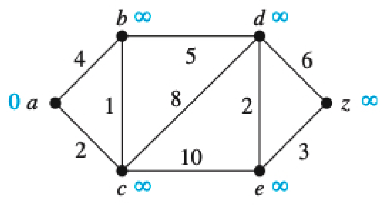


Figure 7.6

EXAMPLE 7.2 A traveling salesperson wants to visit each of n cities exactly once and return to his starting point. For example, suppose that the salesperson wants to visit Detroit, Toledo, Saginaw, Grand Rapids, and Kalamazoo (see Figure (7.7) below). In which order should he visit these cities to travel the minimum total distance?

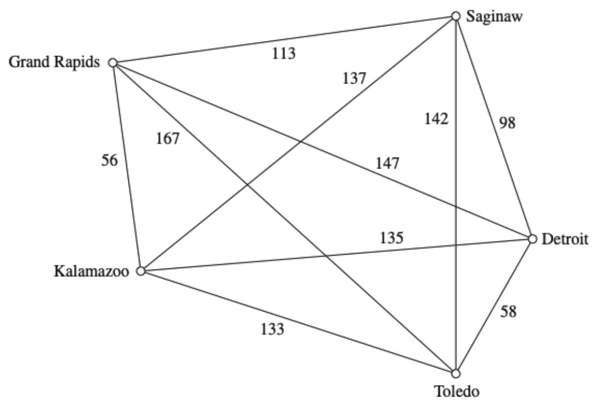


Figure 7.7

^{0}**REFERENCES**

(i) *Discrete Mathematics*, Richard Johnsonbaugh.