Chapter 2
Business Efficiency

For All Practical Purposes: Effective Teaching

- Eye contact is very important in making a connection with students. Your body language along with the intonation of your voice can all attribute to setting a relaxed atmosphere for students to feel comfortable in approaching you or asking questions.
- Make it clear to students early on during the term regarding your policy of asking questions during lecture. If you enjoy having an active dialog during your lecture, convey that to students.

Chapter Briefing

In this chapter, you will be examining *Hamiltonian circuits.* A Hamiltonian circuit contains every vertex of the graph (and only once) but has no known conditions that guarantee its existence. Certain graphs will however have some guaranteed while others will have them excluded. The *traveling salesman problem* (*TSP*) is an optimization problem in order to find the cheapest route while visiting several cities and then return home. Two methods for finding solutions (*nearest-neighbor algorithm* and *sorted-edges algorithm*) to the TSP are discussed in this chapter. Unfortunately, they are not always optimal even though they can be fast. Other methods, such as *brute force*, will lead to the optimal solution but may be very slow.

A connected graph with no circuits, called *trees*, are also discussed in this chapter. A *spanning tree* in a given graph is a tree built using all the vertices of the graph and just enough of its edges to obtain a tree. The *minimum-cost spanning tree* problem is to find a spanning tree of least total edge weight in a given weighted graph. *Kruskal's algorithm* produces an optimal solution to this problem.

The final topic in this chapter deals with *order-requirement digraphs*. The vertices of the graph represent the tasks, and the edges are directed from one vertex to another. Directed edges represent the allowable direction of travel. A certain directed path in this graph, the *critical path*, corresponds to the sequence of tasks that will take the longest time to complete. Since our job is not complete until every possible sequence of tasks has been finished, the "length" of the critical path tells us the least amount of time it will take us to complete our job. It is possible for a digraph to have more than one critical path.

In order to facilitate your preparation, the **Chapter Topics to the Point** has been broken down into **Fundamental Counting Principle**, **Hamiltonian Circuits**, **Finding a Solution to the TSP**, **Trees**, and **Order-requirement digraphs**. Examples with solutions for these topics that do not appear in the text nor study guide are included in the *Teaching Guide*. You should feel free to use these examples in class, if needed.

The last section of this chapter of *The Teaching Guide for the First-Time Instructor* is **Solutions to Student Study Guide ✎ Questions**. These are the complete solutions to the six questions included in the *Student Study Guide*. Students only have the answers to these questions, not the solutions.

Chapter Topics to the Point

Fundamental Counting Principle

The **fundamental principle of counting** says that if there are a ways of choosing one thing, b ways of choosing a second after the first, …, and z ways of choosing the last item after the earlier choices then there are a total of $a \times b \times ... \times z$ total ways to make such choices.

Example
When getting dressed, you can choose from 3 pants, 2 shirts, and 4 belts. How many different outfits can be created?

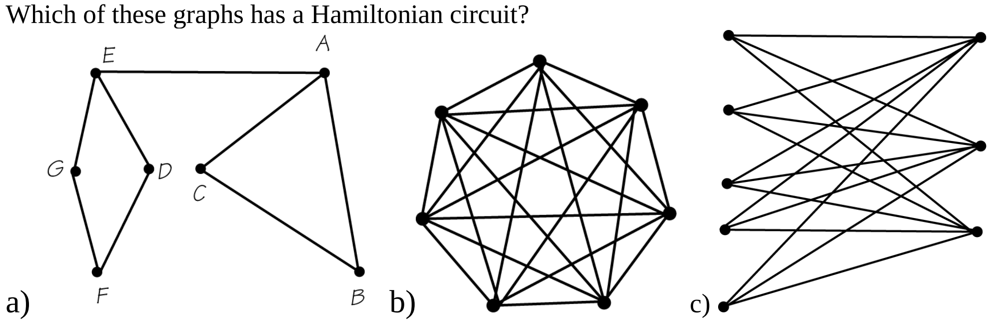
Solution
According to the fundamental principle of counting, there would be $3 \times 2 \times 4 = 24$ different outfits.

Teaching Tip
Demonstrate fundamental counting principle exercises also with tree diagrams in order to lead into finding Hamilton circuits (next section).

Hamiltonian Circuits

A **Hamiltonian circuit** of a graph visits each vertex exactly once, and returns to the starting point. There is no simple way to determine if a graph has a Hamiltonian circuit, and it can be hard to construct one. Graphs may fail to have Hamiltonian circuits for a variety of reasons. One very important class of graphs, the **complete** graphs, automatically have Hamiltonian circuits. A graph is complete if an edge is present between any pair of vertices. If a complete graph has n vertices, then there are $\frac{(n-1)!}{2}$ Hamiltonian circuits.

Example
Which of these graphs has a Hamiltonian circuit?



Solution
a) No, the graph does not have a Hamiltonian circuit. The edge *EA* divides the graph into two parts. If you start the tour at a vertex in one part and then cross *EA*, you cannot get back to the starting vertex without crossing *EA* again.

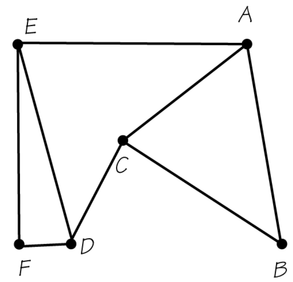
b) This graph is from a family of graphs known to have Hamiltonian circuits – the family of complete graphs; every pair of vertices is joined by an edge. In this case we know there will be $\frac{(7-1)!}{2} = \frac{6!}{2} = \frac{6 \times 5 \times 4 \times 3 \times 2 \times 1}{2} = \frac{720}{2} = 360$ distinct circuits.

c) This graph is from a family of graphs known not to have Hamiltonian circuits: it was constructed with vertices on two parallel vertical columns with one column having more vertices than the other.

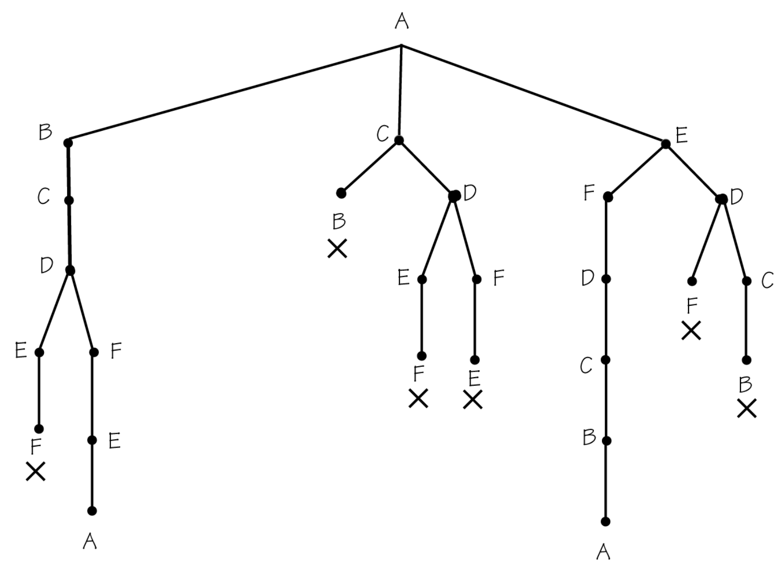
The **method of trees** can be used to help generate all possible Hamiltonian circuits.

Example

Use the method of trees to find all possible Hamiltonian circuits for this graph starting at *A*.



Solution



There is only one distinct Hamiltonian circuit: *ABCDFEA*. Its reversal is *AEFDCBA*.

☝Teaching Tip

Note that the reversals should always be present in the tree diagram. Also point out any kind of symmetry or patterns that may appear in the tree diagrams when compared to the original graph.

Finding a Solution to the TSP

A number added to the edge of a graph is called a **weight**. A **minimum-cost Hamiltonian circuit** is one with the lowest possible sum of the weights of its edges. The problem of finding this minimum-cost Hamiltonian circuit is called the **traveling salesman problem (TSP)**. It is a common goal in the practice of operations research.

The **brute force method** is one algorithm that can be used to find a minimum-cost Hamiltonian circuit, but it is not a very practical method for a large problem because all possibilities must be tried.

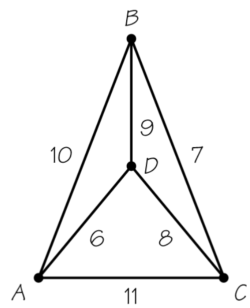
We use a variety of **heuristic** (or "fast") **algorithms** to find solutions to the TSP. Some are very good, even though they may not be optimal. Heuristic algorithms come close enough to giving optimal solutions to be important for practical use.

The **nearest-neighbor algorithm** repeatedly selects the closest neighboring vertex not yet visited in the circuit (with a choice of edges, choose the one with the smallest weight), and returns to the initial vertex at the end of the tour.

The **sorted-edges algorithm** (which, like nearest neighbor, is a **greedy algorithm**) is another heuristic algorithm that can lead to a solution that is close to optimal.

Example

Consider the following graph.



a) Determine the minimum-cost Hamilton circuit using the brute force method starting at *A*.

b) What cost is determined by the nearest-neighbor algorithm starting at *C*?

c) What cost is determined by the sorted-edges algorithm?

Solution

a) There are three distinct Hamilton circuits in the graph starting at *A*.

Hamilton circuit	Cost
ABDCA (*ACDBA*)	10 + 9 + 8 + 11 = 38
ADCBA (*ABCDA*)	**6 + 8 + 7 + 10= 31**
ADBCA (*ACBDA*)	6 + 9 + 7 + 11 = 33

The minimum cost is 31.

b) From *C*, the closest neighbor who is cheapest is *B*. From *B*, the closest neighbor who is cheapest is *D*. Thus, we must go to *A* next, followed by *C* to make a circuit. The total of the weights of the edges in the circuit *CBDAC* is 7 + 9 + 6 + 11= 33.

c) The edges *AD*, *BC*, and *DC* are the cheapest. Although *BD* is next in line to consider, it would form a circuit which doesn't include all vertices. Thus, *AC* is the next edge considered and the complete circuit starting at *A* would be *ADCBA* for cost of 31.

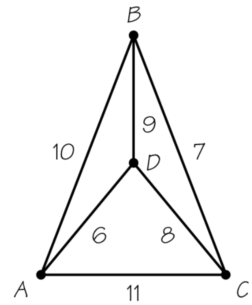
Teaching Tip

Ask students to think about the nearest-neighbor algorithm and what would occur if one were to consider finding the Hamiltonian circuit given by this method for starting at all vertices. Notice in the last example, part b would yield a different result if we started at *A*.

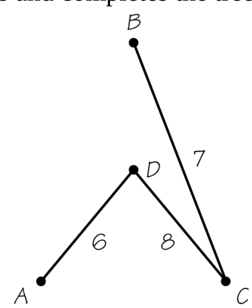
Trees

A **tree** will consist of one piece and contains no circuits. A **spanning tree** is a tree that connects all vertices of a graph to each other with no redundancy (e.g., for a communications network.) A **minimum-cost spanning tree** is most economical. **Kruskal's algorithm** produces one quickly. This algorithm adds links together in order of cheapest cost so that no circuits form and so that every vertex belongs to some link added.

Example
Using this graph again, apply Kruskal's algorithm to find a minimum-cost spanning tree.



Solution
AD, *BC*, and *DC* are the cheapest edges and completes the tree.



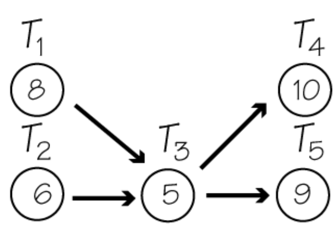
☝**Teaching Tip**
Emphasize that trees do not form circuits.

Order-Requirement Digraph

The necessary arrangement of tasks in a complex job can be represented in an **order-requirement digraph** (directed graph), with arrows showing the order requirements.

The longest path in an order-requirement digraph is called the **critical path**. The length is measured in terms of summing the task times of the tasks making up the path. The length of the longest path corresponds to the earliest completion time.

Example
Find the earliest completion time for this ordered-requirement digraph.

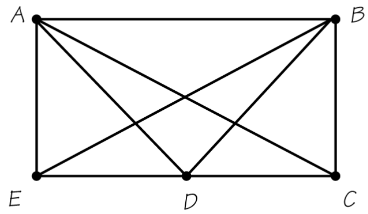


Solution
The earliest completion time corresponds to the length of the critical path. The critical path (longest path) would be $T_1T_3T_4$. The earliest completion time is $8+5+10=23$.

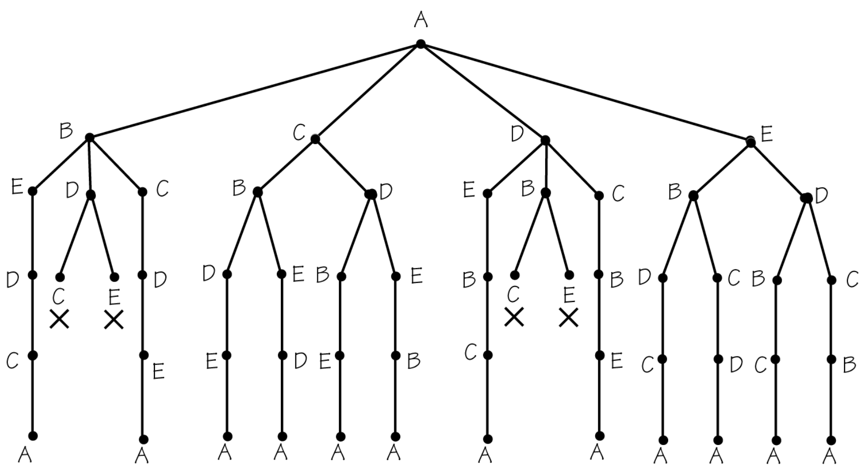
Solutions to Student Study Guide ✎ Questions

Question 1

Use the method of trees to determine the number of distinct Hamiltonian circuits for this graph starting at *A*.



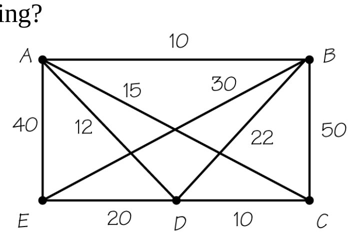
Solution



ABEDCA (*ACDEBA*), *ABCDEA* (*AEDCBA*), *ACBDEA* (*AEDBCA*), *ACBEDA* (*ADEBCA*), *ACDBEA* (*AEBDCA*), and *ADCBEA* (*AEBCDA*) yield 6 distinct Hamilton circuits.

Question 2

What is the minimum cost when using the brute force method to determine the minimum-cost Hamiltonian circuit for the following?



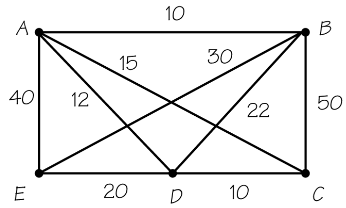
Solution

Hamilton circuit	Cost
ABEDCA	**10 + 30 + 20 + 10 + 15 = 85**
ABCDEA	10 + 50 + 10 + 20 + 40 = 130
ACBDEA	15 + 50 + 22 + 20 + 40 = 147
ACBEDA	15 + 50 + 30 + 20 + 12 = 127
ACDBEA	15 + 10 + 22 + 30 + 40 = 117
ADCBEA	12 + 10 + 50 + 30 + 40 = 142

The minimum cost is 85.

Question 3

Use the nearest-neighbor algorithm to find a Hamiltonian circuit for this graph starting at *C*. What is the total weight?

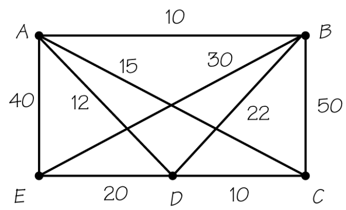


Solution

From *C*, the closest neighbor who is cheapest is *D*. From *D*, the closest neighbor who is cheapest is *A*. Now although going to *B* next would be cheapest, it would force us to revisit a vertex in order to get back to *C*. Thus, we must go to *E* next, the *B* followed by *C* to make a circuit. The total of the weights of the edges in the circuit *CDAEBC* is 10 + 12 + 40 + 30 + 50 = 142.

Question 4

Use the sorted-edges algorithm to find a Hamiltonian circuit for this graph. What is the total weight?

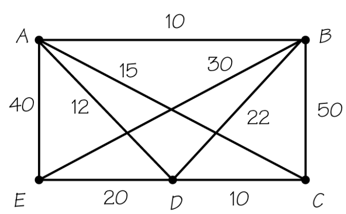


Solution

The edges *AB* and *CD* are the cheapest. Although *AD* is next in line to consider, it cannot be used with *AB* and *CD* in a circuit. Thus, *AC* is the next edge considered and it can make a circuit by including *BE* and *ED*. Thus a complete circuit starting at *A* would be *ABEDCA* for cost of 85.

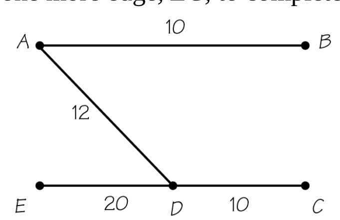
Question 5

Using this graph again, apply Kruskal's algorithm to find a minimum-cost spanning tree. What is the minimum cost?



Solution

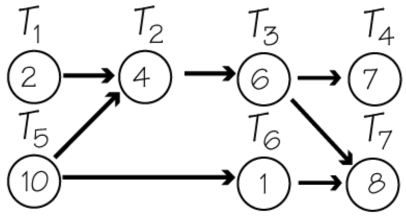
AB, *CD*, and *AD* are the cheapest edges and are chosen first. The next cheapest one, *AC*, would close a loop. In any case, we only need one more edge, *EC*, to complete the tree.



The cost is 10 + 12 + 20 + 10 =52.

Question 6

What is the length of the earliest completion time for this ordered-requirement digraph?



Answer

$T_5 T_2 T_3 T_7$ is the path through the digraph with the greatest total time is $10+4+6+8=28.$ This total time represents the earliest completion time.