

EXAMPLE 13 Complete Bipartite Graphs A **complete bipartite graph** $K_{m,n}$ is a graph that has its vertex set partitioned into two subsets of m and n vertices, respectively with an edge between two vertices if and only if one vertex is in the first subset and the other vertex is in the second subset. The complete bipartite graphs $K_{2,3}$, $K_{3,3}$, $K_{3,5}$, and $K_{2,6}$ are displayed in Figure 9. ◀

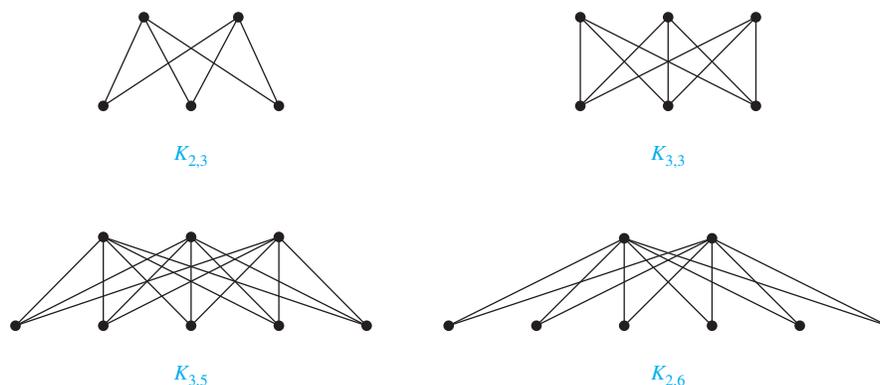


FIGURE 9 Some Complete Bipartite Graphs.

Bipartite Graphs and Matchings

Bipartite graphs can be used to model many types of applications that involve matching the elements of one set to elements of another, as Example 14 illustrates.

EXAMPLE 14 Job Assignments Suppose that there are m employees in a group and n different jobs that need to be done, where $m \geq n$. Each employee is trained to do one or more of these n jobs. We would like to assign an employee to each job. To help with this task, we can use a graph to model employee capabilities. We represent each employee by a vertex and each job by a vertex. For each employee, we include an edge from that employee to all jobs that the employee has been trained to do. Note that the vertex set of this graph can be partitioned into two disjoint sets, the set of employees and the set of jobs, and each edge connects an employee to a job. Consequently, this graph is bipartite, where the bipartition is (E, J) where E is the set of employees and J is the set of jobs. We now consider two different scenarios.

First, suppose that a group has four employees: Alvarez, Berkowitz, Chen, and Davis; and suppose that four jobs need to be done to complete Project 1: requirements, architecture, implementation, and testing. Suppose that Alvarez has been trained to do requirements and testing; Berkowitz has been trained to do architecture, implementation, and testing; Chen has been trained to do requirements, architecture, and implementation; and Davis has only been trained to do requirements. We model these employee capabilities using the bipartite graph in Figure 10(a).

Second, suppose that a group has second group also has four employees: Washington, Xuan, Ybarra, and Ziegler; and suppose that the same four jobs need to be done to complete Project 2 as are needed to complete Project 1. Suppose that Washington has been trained to do architecture; Xuan has been trained to do requirements, implementation, and testing; Ybarra has been trained to do architecture; and Ziegler has been trained to do requirements, architecture and testing. We model these employee capabilities using the bipartite graph in Figure 10(b).

To complete Project 1, we must assign an employee to each job so that every job has an employee assigned to it, and so that no employee is assigned more than one job. We can do this by assigning Alvarez to testing, Berkowitz to implementation, Chen to architecture, and Davis to requirements, as shown in Figure 10(a) (where blue lines show this assignment of jobs).

To complete Project 2, we must also assign an employee to each job so that every job has an employee assigned to it and no employee is assigned more than one job. However, this is