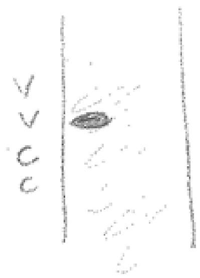
Section 1.4 – The Language of Graphs

The following is a variation of a famous puzzle often used as an example in the study of artificial intelligence. It concerns an island on which all the people are of one of two types, either vegetarians or cannibals. Initially, two vegetarians and two cannibals are on the left bank of a river. With them is a boat that can hold a maximum of two people. The aim of the puzzle is to find a way to transport all the vegetarians and cannibals to the right bank of the river. What makes this difficult is that at no time can the number of cannibals on either bank outnumber the number of vegetarians. Otherwise, disaster befalls the vegetarians!

A picture might help:



Partner up

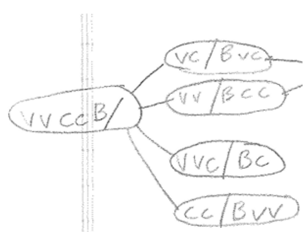
Find a sequence where everyone can be transported to the other side safely.

Share.

The next question is more difficult:
How many different ways are there for everyone to be transported to the other side safely?

A systematic way to approach this problem is to introduce a notation that can indicate all possible arrangements of vegetarians, cannibals, and the boat on the banks of the river. For example, you could write $(vvc\,/\,Bc)$ to indicate that there are two vegetarians and one cannibal on the left bank and one cannibal and the boat on the right bank. Then $(vvccB\,/)$ would indicate the initial position in which both vegetarians, both cannibals, and the boat are on the left bank of the river. The aim of the puzzle is to figure out a sequence of moves to reach the position $(/\,Bvvcc)$ in which both vegetarians, both cannibals, and the boat are on the right bank of the river.

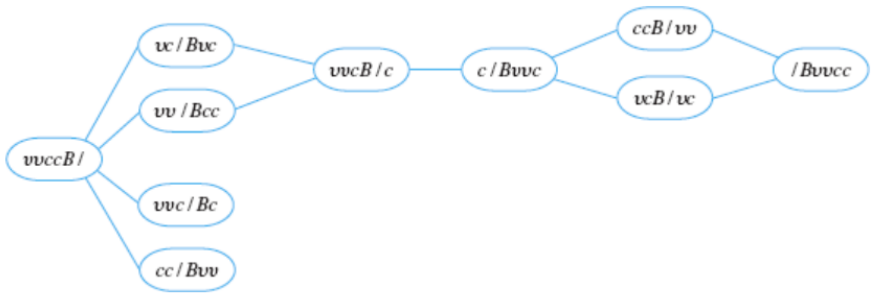
Create a graph. A graph has vertices and edges. With a starting state of ***v v c c B /***, give all the possible following states:



Work through the rest of the possibilities. Now, can you answer the total number of successful transports?

edges from one vertex to another. The rationale for drawing undirected edges is that each legal move is reversible.) From the position (vc / Bvc), the only legal moves are to go back to ($vvccB$ /) or to go to ($vvcB$ / c). You can also show these by drawing in edges. Continue this process until finally you reach (/ $Bvvcc$). From Figure 1.4.3 it is apparent that one successful sequence of moves is ($vvccB$ /) $\rightarrow$ (vc / Bvc) $\rightarrow$ ($vvcB$ / c) $\rightarrow$ (c/ $Bvvc$) $\rightarrow$ (ccB / vv) $\rightarrow$ (/ $Bvvcc$).

Figure 1.4.3



Partner up

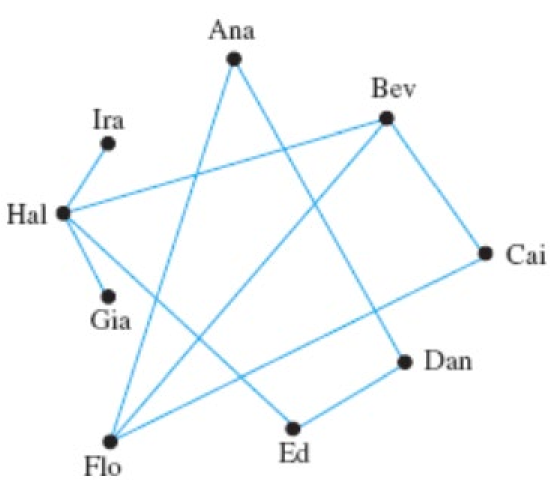
Name	Previous Partners
Ana	Dan, Flo
Bev	Cai, Flo, Hal
Cai	Bev, Flo
Dan	Ana, Ed
Ed	Dan, Hal
Flo	Cai, Bev, Ana
Gia	Hal
Hal	Gia, Ed, Bev, Ira
Ira	Hal

The goal is to set up teams of three to work on some projects.

In addition, the goal is to maximize the number of people who had previously working together.

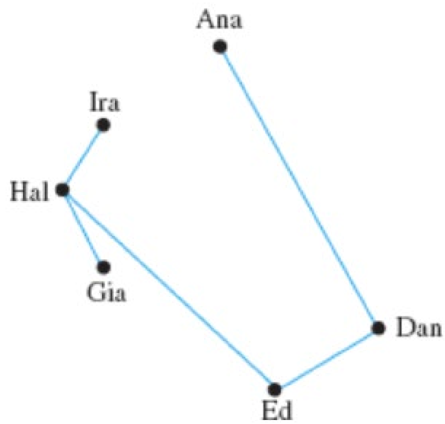
Draw a graph representing the given information.

==



How can you use this graph to assign teams of three?

It's easy to see that Bev, Cai, and Flo worked together (there's a triangle). So put them on a team, and remove them from the graph.



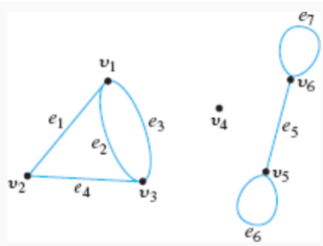
So, how about:

Bev/Cai/Flo

Hal/Ira/Gia

Ana/Dan/Ed

Everyone would have at least one other person on the team they've worked with before.



a. Write the vertex set and the edge set, and give a table showing the edge-endpoint function.

b. Find all edges that are incident on v_1, all vertices that are adjacent to v_1, all edges that are adjacent to e_1, all loops, all parallel edges, all vertices that are adjacent to themselves, and all isolated vertices.

===

a. vertex set $= \{v_1, v_2, v_3, v_4, v_5, v_6\}$

edge set $= \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$

Edge	Endpoints
e_1	$\{v_1, v_2\}$
e_2	$\{v_1, v_3\}$
e_3	$\{v_1, v_3\}$
e_4	$\{v_2, v_3\}$
e_5	$\{v_5, v_6\}$
e_6	$\{v_5\}$
e_7	$\{v_6\}$

Note that the isolated vertex does not appear in the Edge-Endpoint table, since a vertex doesn't need to be an endpoint of an edge.

Also, the Edge-Endpoint table is what a defines a graph, not how the graph is drawn. There are many ways to draw the same graph.

b. e_1, e_2, and e_3 are incident on v_1.

v_2 and v_3 are adjacent to v_1.

e_2, e_3, and e_4 are adjacent to e_1.

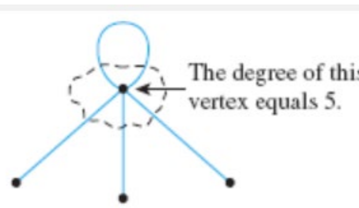
e_6 and e_7 are loops.

e_2 and e_3 are parallel.

v_5 and v_6 are adjacent to themselves.

v_4 is an isolated vertex.

A **directed graph**, or **digraph**, consists of two finite sets: a nonempty set $V(G)$ of vertices and a set $D(G)$ of directed edges, where each is associated with an ordered pair of vertices called its **endpoints**. If edge e is associated with the pair (v, w) of vertices, then e is said to be the (**directed**) **edge** from v to w.



Let G be a graph and v a vertex of G. The **degree of** v, denoted $\mathbf{deg}(v)$, equals the number of edges that are incident on v, with an edge that is a loop counted twice.