Mathematical concepts for computer science

Graphs

A graph **G = (V ,E)** consists of V , a nonempty set of vertices (or nodes) and E, a set of edges.

Each edge has either one or two vertices associated with it, called its **endpoints**.

An edge is said to **connect** its endpoints.

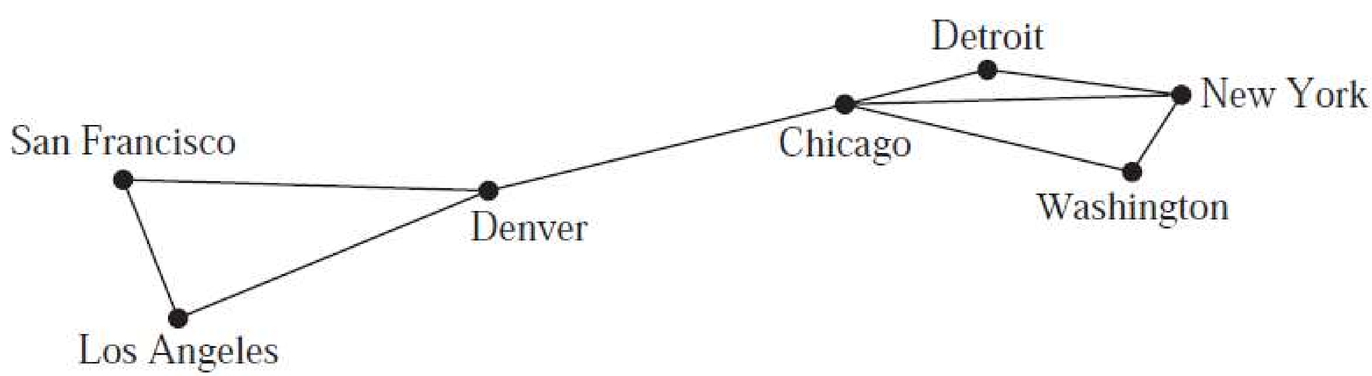


FIGURE 1 **A Computer Network.**

Simple graph

- A graph in which **each edge connects two different vertices** and where **no two edges connect the same pair of vertices** is called a **simple graph.**

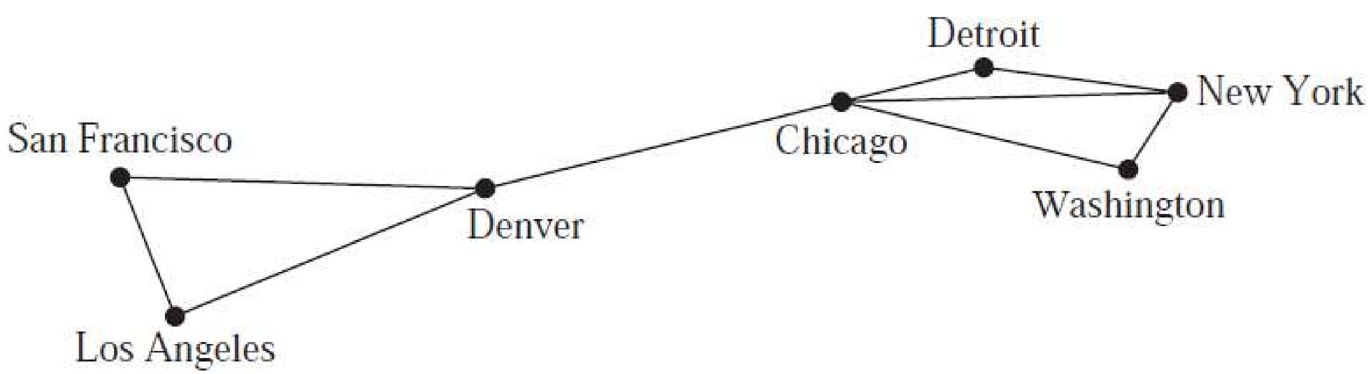


FIGURE 1 **A Computer Network.**

Whether the above graph is a simple graph ?

Simple graph

- A graph in which **each edge connects two different vertices** and where **no two edges connect the same pair of vertices** is called a **simple graph.**

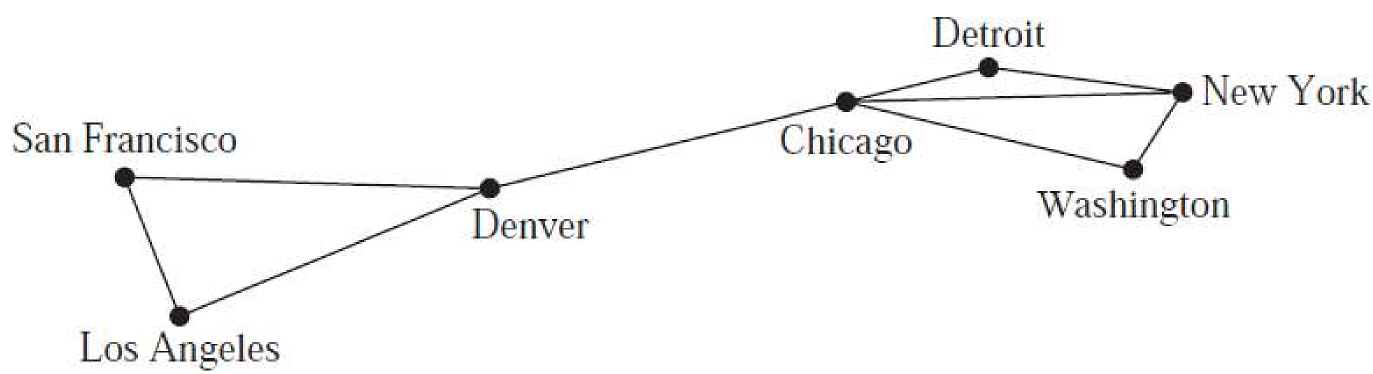


FIGURE 1 **A Computer Network.**

Yes, This is a simple graph

Multigraphs

- Graphs that may have **multiple edges connecting the same vertices are called multigraphs.**
- Eg: A computer network may contain multiple links between data centers.

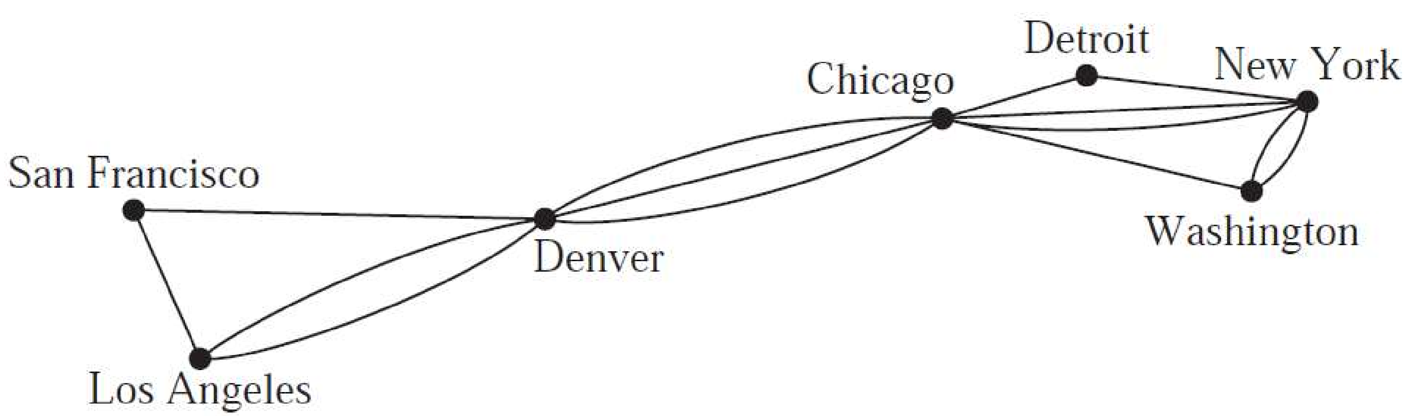


FIGURE 2 **A Computer Network with Multiple Links between Data Centers.**

Directed graphs

- In a computer network, some links may operate in only one direction.
- A **directed graph** (or **digraph**) (V ,E) consists of a **nonempty set of vertices V** and **a set of directed edges** (or arcs) **E**.
- Each directed edge is associated with an **ordered pair of vertices**.
- The directed edge associated with the ordered pair **(u, v)** is said to **start at u and end at v**.

Simple directed graph

- When a directed graph has no loops and has no multiple directed edges, it is called a **simple directed graph.**

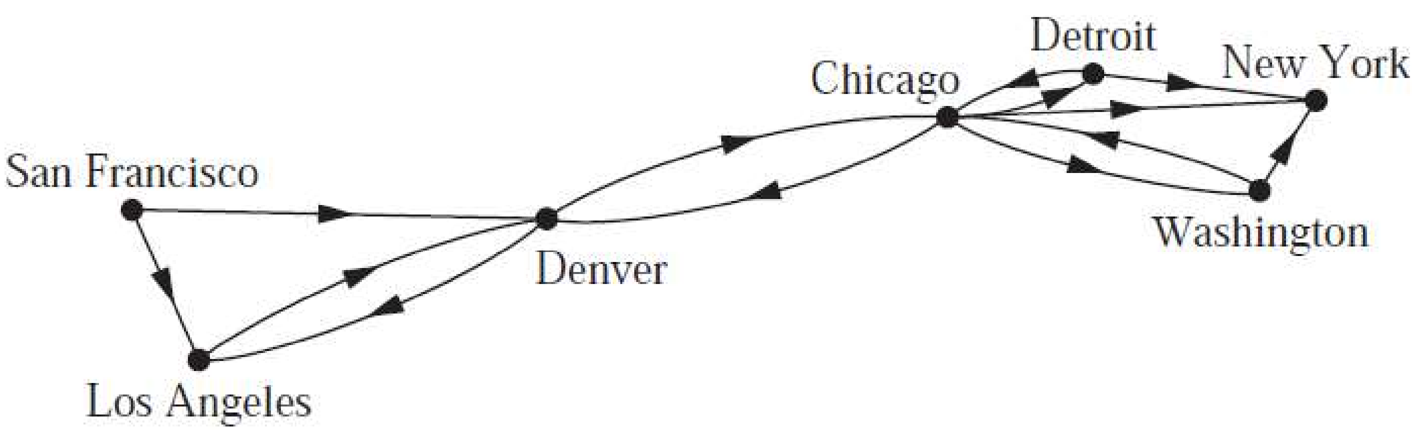


FIGURE 4 A Communications Network with One-Way Communications Links.

Directed multigraphs

- Directed graphs that may have **multiple directed edges** from a vertex to a second (possibly the same) vertex are used to model such networks. We called such graphs **directed multigraphs.**
- When there are **m** directed **edges**, each associated to an ordered pair of **vertices (u, v)**, we say that **(u, v) is** an edge of **multiplicity m.**

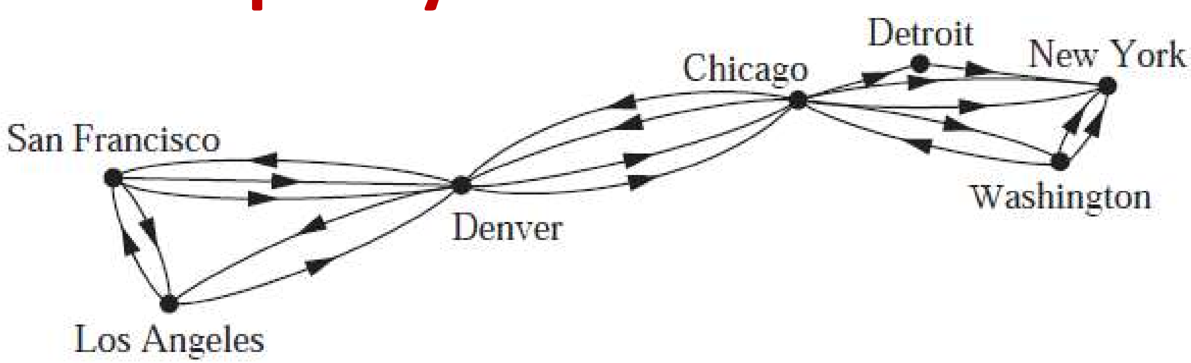


FIGURE 5 A Computer Network with Multiple One-Way Links.

Mixed graph

- A graph with both directed and undirected edges is called a **mixed graph.**
- For example, a mixed graph might be used to model a computer network containing links that operate in both directions and other links that operate only in one direction.

Graph Terminology

TABLE 1 **Graph Terminology.**

Type	*Edges*	*Multiple Edges Allowed?*	*Loops Allowed?*
Simple graph	Undirected	No	No
Multigraph	Undirected	Yes	No
Pseudograph	Undirected	Yes	Yes
Simple directed graph	Directed	No	No
Directed multigraph	Directed	Yes	Yes
Mixed graph	Directed and undirected	Yes	Yes

Adjacent vertices

- Two vertices u and v in an undirected graph G are called **adjacent** (or neighbors) in G if u and v are endpoints of an edge e of G. Such an edge **e** is called **incident** with the vertices u and v and **e** is said to **connect** u and v.

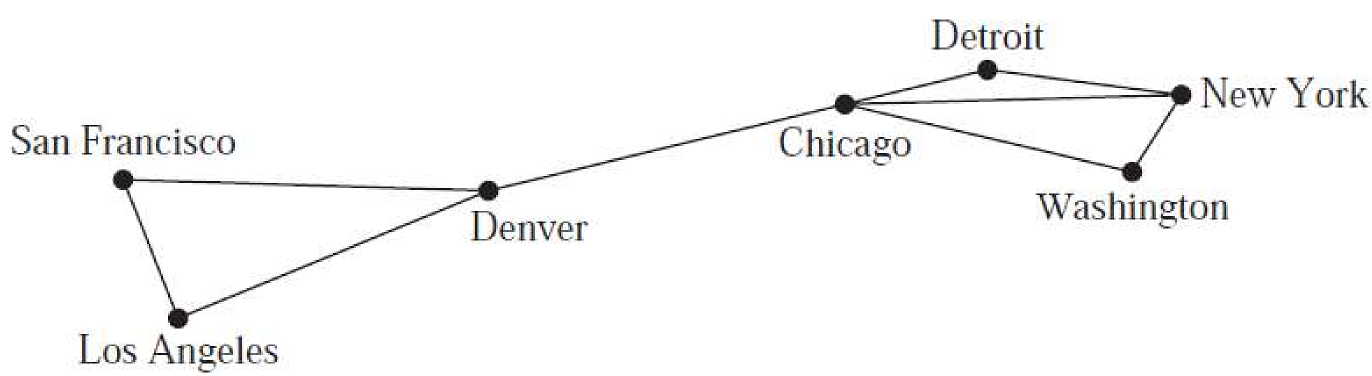


FIGURE 1 **A Computer Network.**

Neighborhood

- The set of all neighbors of a vertex v of G = (V ,E), denoted by **N(v)**, is called the neighborhood of v. **If A is a subset of V** , we denote by N(A) the set of all vertices in G that are **adjacent to at least one vertex in A**. So, $N(A) = \bigcup_{v \in A} N(v)$.

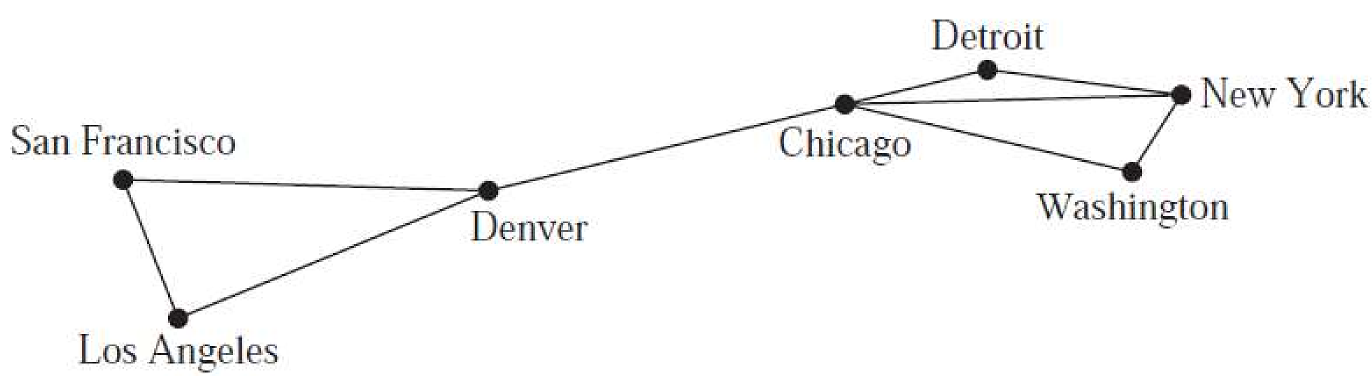
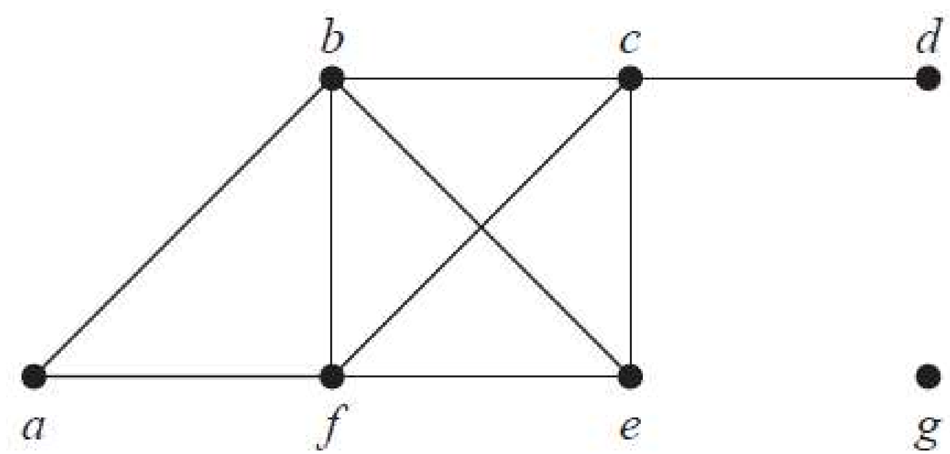


FIGURE 1 **A Computer Network.**

Degree of a vertex

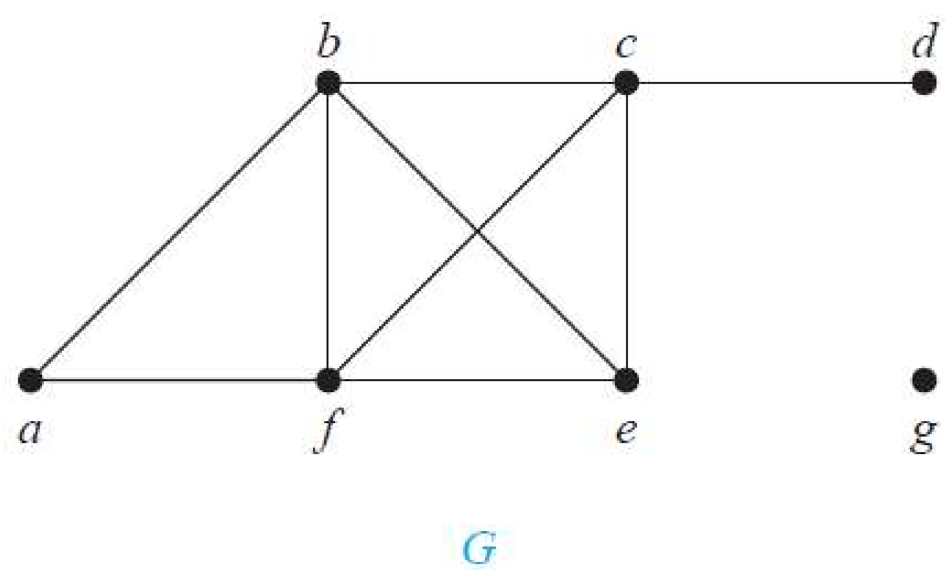
The **degree of a vertex** in an **undirected graph** is the **number of edges incident with it**, **except that a loop** at a vertex contributes twice to the degree of that vertex. The degree of the vertex v is denoted by deg(v).



G

Degree of a vertex

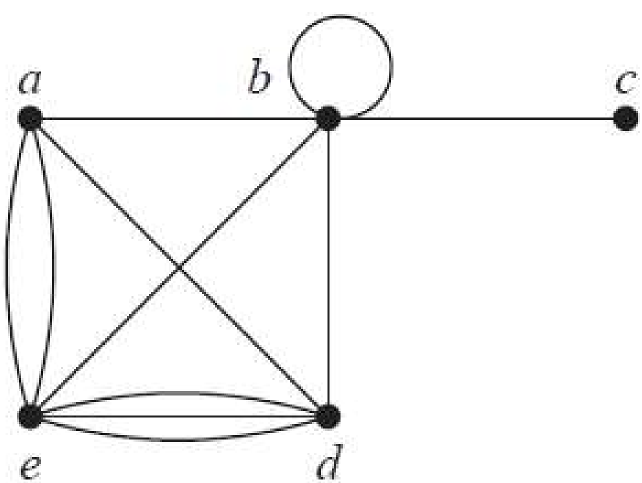
The **degree of a vertex** in an **undirected graph** is the **number of edges incident with it**, **except that a loop** at a vertex contributes twice to the degree of that vertex. The degree of the vertex v is denoted by deg(v).



deg(a) = 2
deg(b) = deg(c) = deg(f) = 4
deg(d) = 1
deg(e) = 3
deg(g) = 0.

Degree of a vertex

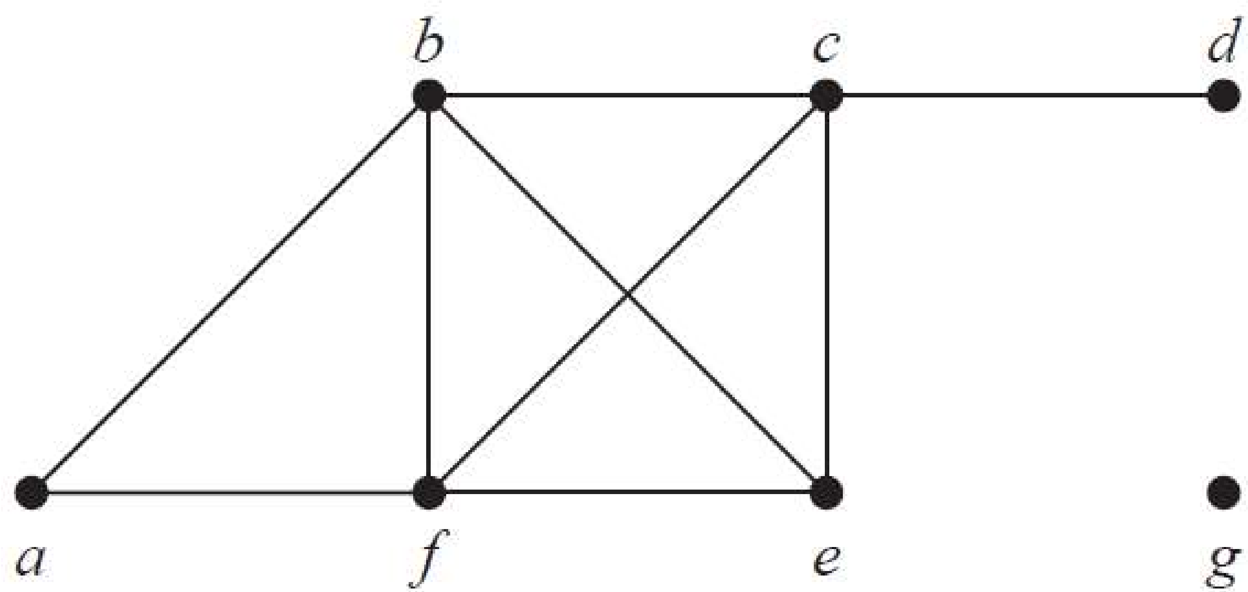
The **degree of a vertex** in an **undirected graph** is the **number of edges incident with it**, **except that a loop at a vertex contributes twice** to the degree of that vertex. The degree of the vertex v is denoted by deg(v).



H

Isolated and Pendant vertex

- A vertex of degree zero is called **isolated.**
  - **an isolated vertex is not adjacent** to any vertex.
- A vertex is **pendant if and only if** it has degree one.
  - A pendant vertex is adjacent to exactly one other vertex.



THE HANDSHAKING THEOREM

Let G = (V ,E) be an undirected graph with m edges.

$$\text{Then } 2m = \sum_{v \in V} \deg(v)$$

(Note that this applies even if multiple edges and loops are present.)

How many edges are there in a graph with 10 vertices each of degree six?

THE HANDSHAKING THEOREM

Let G = (V ,E) be an undirected graph with m edges.

$$\text{Then } 2m = \sum_{v \in V} \deg(v)$$

(Note that this applies even if multiple edges and loops are present.)

How many edges are there in a graph with 10 vertices each of degree six?

- Because the sum of the degrees of the vertices is **6 * 10 = 60**,
- it follows that **2m = 60** where m is the number of edges. Therefore, **m = 30**.

THE HANDSHAKING THEOREM

Let G = (V ,E) be an undirected graph with m edges.

$$\text{Then } 2m = \sum_{v \in V} \deg(v)$$

(Note that this applies even if multiple edges and loops are present.)

Theorem shows that **the sum of the degrees of the vertices of an undirected graph is even**.

THEOREM

An undirected graph has an even number of vertices of odd degree.

Proof:

- Let **V1** and **V2** be the set of vertices of even degree and the set of vertices of odd degree, respectively, in an undirected graph G = (V ,E) with **m** edges.
- Then

$$2m = \sum_{v \in V} \deg(v) = \sum_{v \in V_1} \deg(v) + \sum_{v \in V_2} \deg(v).$$

THEOREM

An undirected graph has an even number of vertices of odd degree.

Proof:

$$2m = \sum_{v \in V} \deg(v) = \sum_{v \in V_1} \deg(v) + \sum_{v \in V_2} \deg(v).$$

even even

Because all the terms in this sum are odd, there must be an even number of such terms.

Adjacent vertices

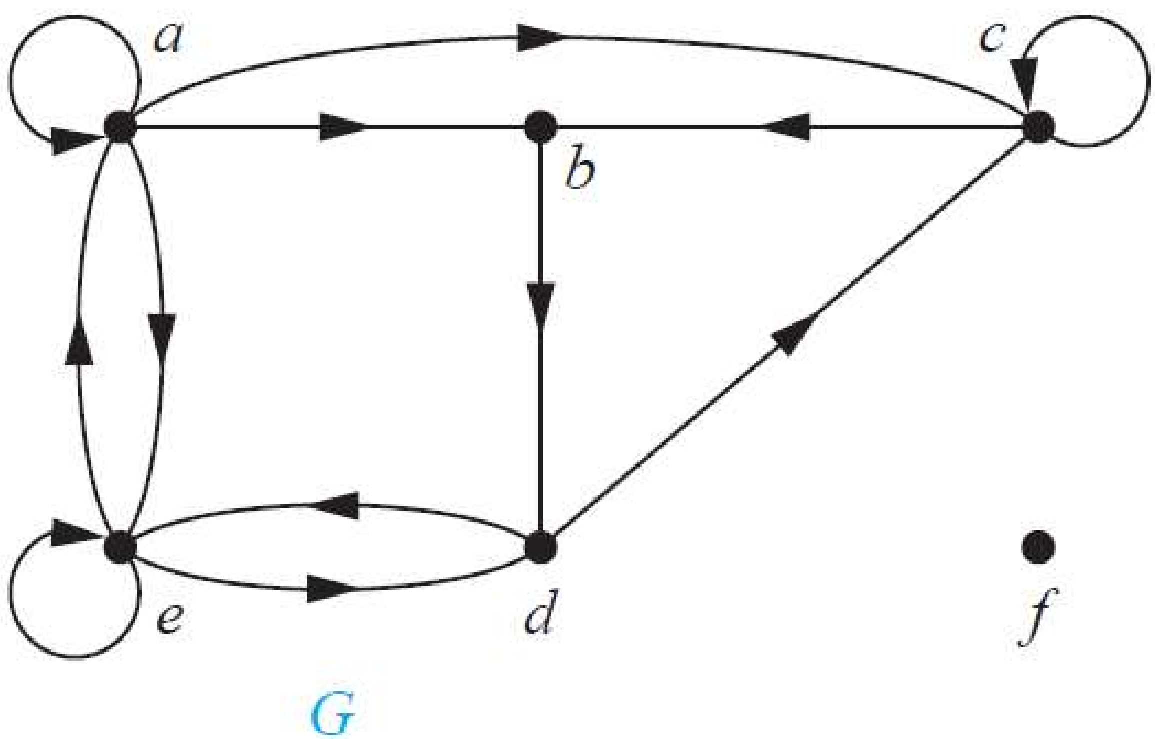
- When **(u, v) is an edge of the graph G** with **directed** edges, **u is said to be adjacent to v** and **v is said to be adjacent from u**.
- The **vertex u** is called the **initial vertex** of (u, v), and **v** is called the **terminal or end vertex** of (u, v).
- The **initial vertex and terminal vertex** of a **loop** are the **same**.

In-degree and out-degree

- In a graph with directed edges the **in-degree** of a vertex v, denoted by **$deg^{-}(v)$**, is the **number of edges with v as their terminal vertex**.
- The **out-degree** of v, denoted by **deg+(v)**, is **the number of edges with v as their initial vertex**.

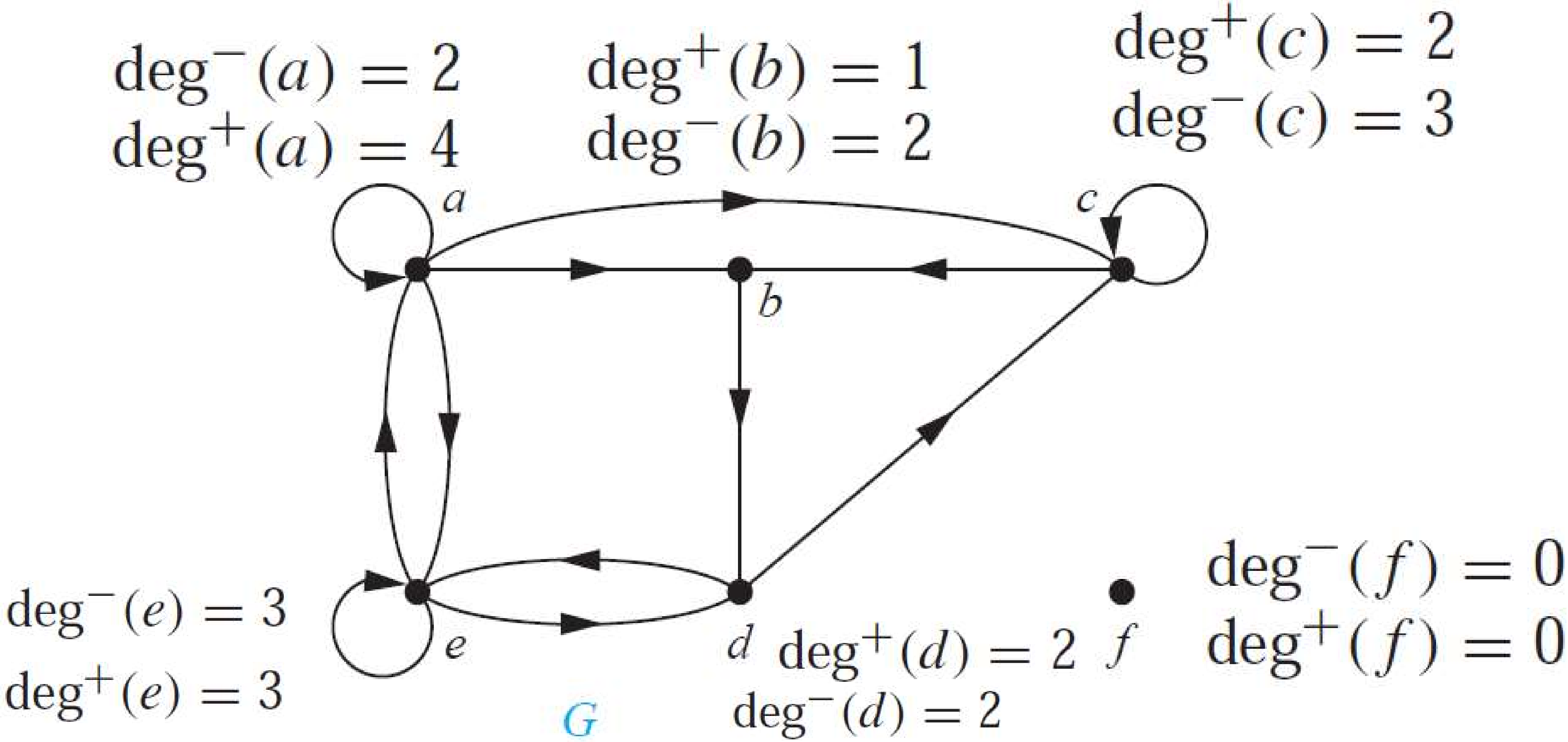
In-degree and out-degree

- Find the in-degree and out-degree of each vertex in the graph G



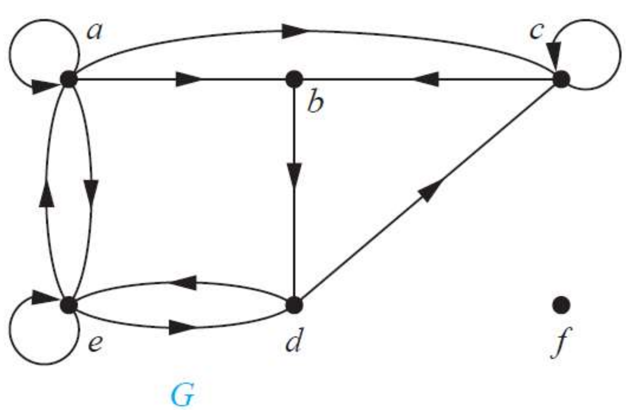
In-degree and out-degree

- Find the in-degree and out-degree of each vertex in the graph G



In-degree and out-degree

- Find the in-degree and out-degree of each vertex in the graph G



$$\sum \deg^- = 12$$

$$\sum \deg^+ = 12$$

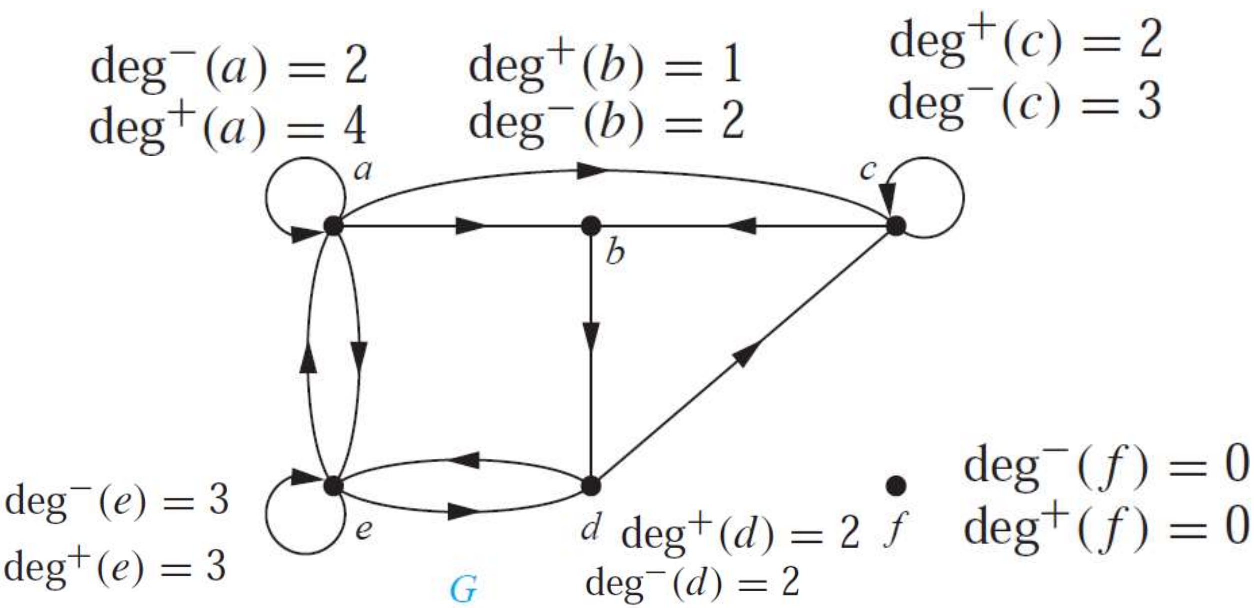
$$\text{number of edges} = 12$$

Because **each edge** has an **initial vertex** and a **terminal vertex**, the **sum of the in-degrees** and the **sum of the out-degrees of all vertices** in a graph **with directed edges** are the **same**.
Both of these sums are the **number of edges** in the graph.

THEOREM

- Let G = (V ,E) be a graph with directed edges. Then

$$\sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v) = |E|$$



$\sum \deg^- = 12$

$\sum \deg^+ = 12$

number of edges $= 12$

Complete Graphs

- A **complete graph** on n vertices, denoted by **K_n**, is a simple graph that contains **exactly one edge between each pair of distinct vertices**.

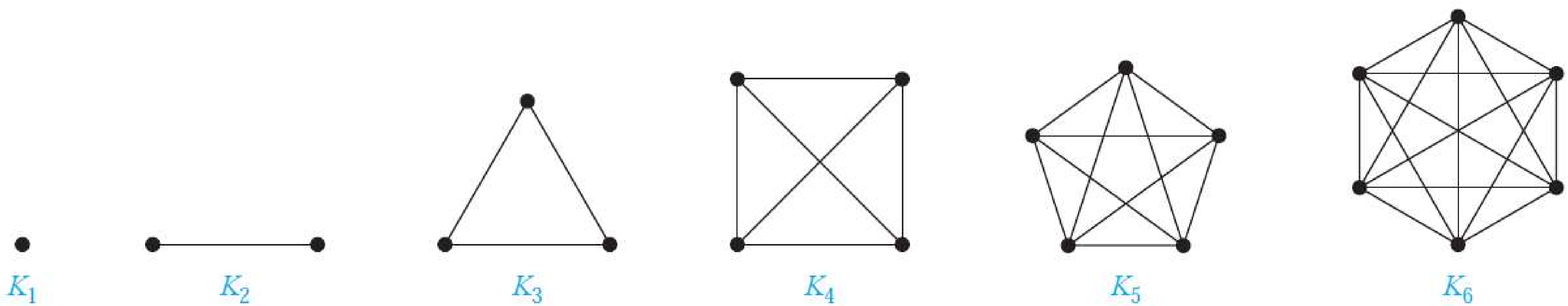


FIGURE 3 The Graphs K_n for $1 \leq n \leq 6$.

Cycles

A **cycle C_n**, $n \geq 3$, consists of

n vertices $v_1, v_2, v_3, \ldots\ldots v_n$

and edges **$\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}$, and $\{v_n, v_1\}$**.

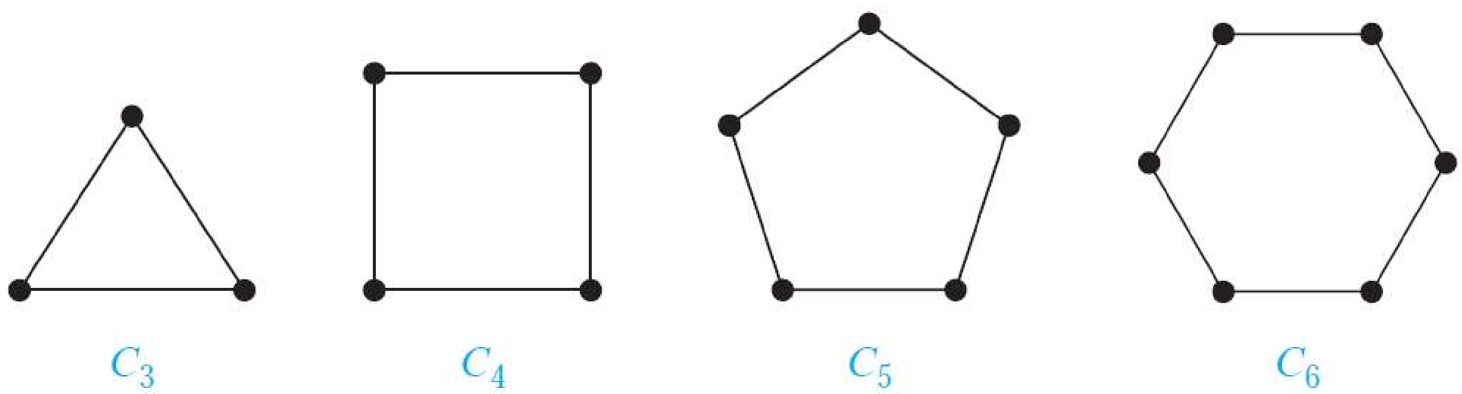


FIGURE 4 The Cycles C_3, C_4, C_5, and C_6.

Wheels

Add an additional vertex to a cycle C_n, for $n \geq 3$, and **connect this new vertex to each of the n vertices in C_n**, by new edges.

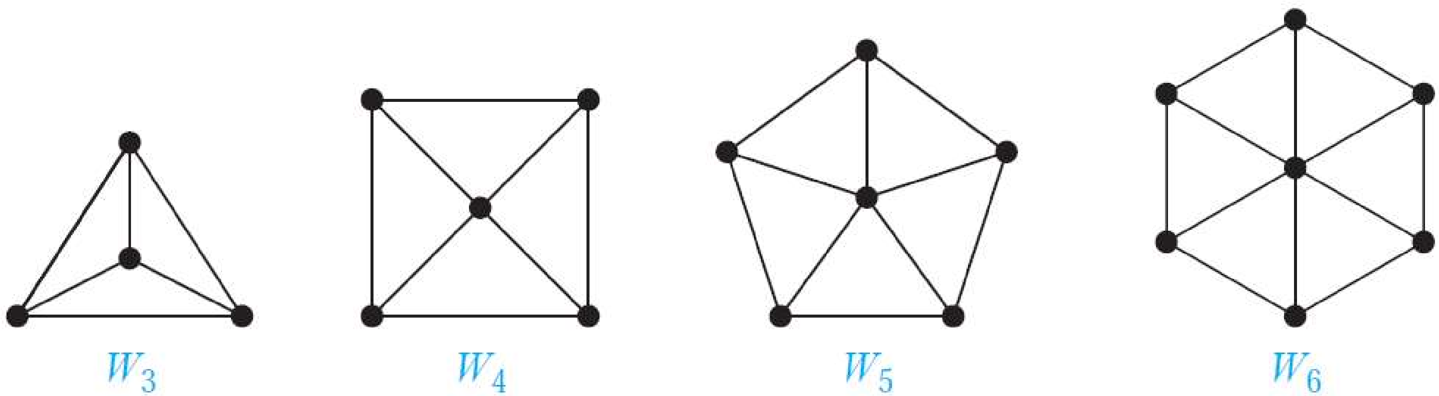


FIGURE 5 The Wheels W_3, W_4, W_5, and W_6.

n-Cubes

- An **n-dimensional hypercube**, or **n-cube**, denoted by **Q_n**, is a graph that has vertices representing the **2^n** bit strings of length **n**
- **Two vertices are adjacent** if and only if the **bit strings** that they represent **differ in exactly one bit position**

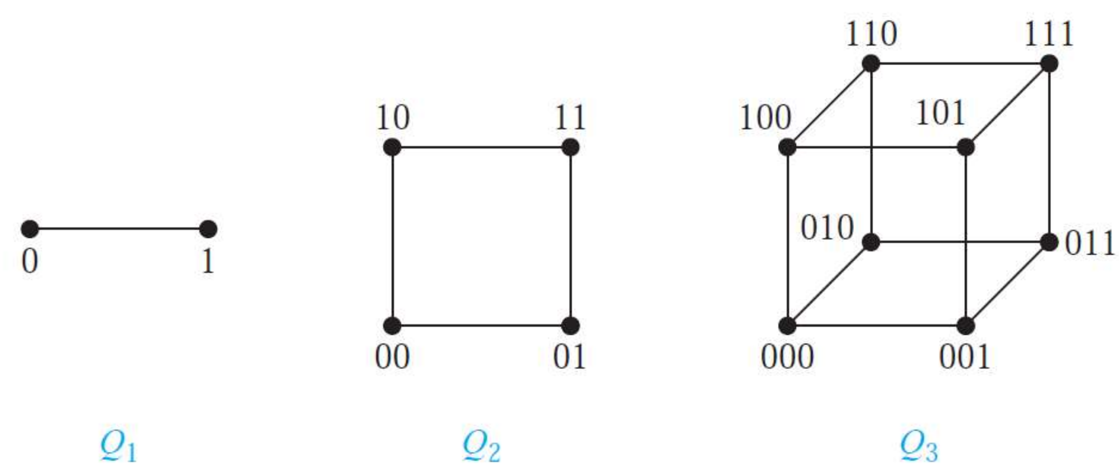


FIGURE 6 The n-cube Q_n, $n = 1, 2, 3$.

Bipartite Graphs

- A simple graph G is called **bipartite** if its vertex **set V can be partitioned into two disjoint sets V1 and V2** such that **every edge in the graph connects a vertex in V1 and a vertex in V2**

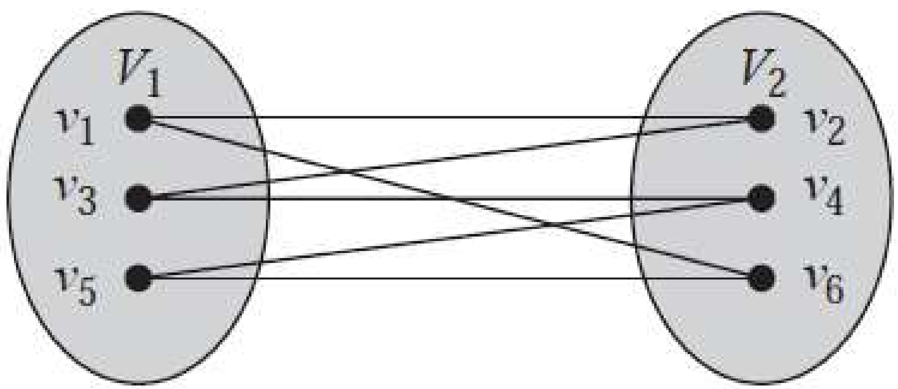
(so that **no edge in G connects either two vertices in V1 or two vertices in V2**).

When this condition holds, we call the pair **(V1, V2) a bipartition of the vertex set V of G**.

Bipartite Graphs

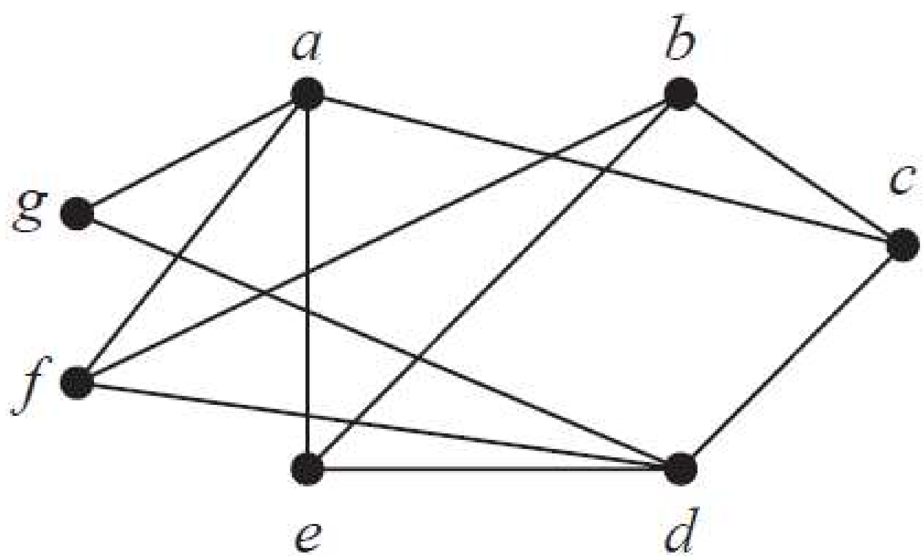
- every edge in the graph connects a vertex in V1 and a vertex in V2

(no edge in G connects either two vertices in V1 or two vertices in V2).



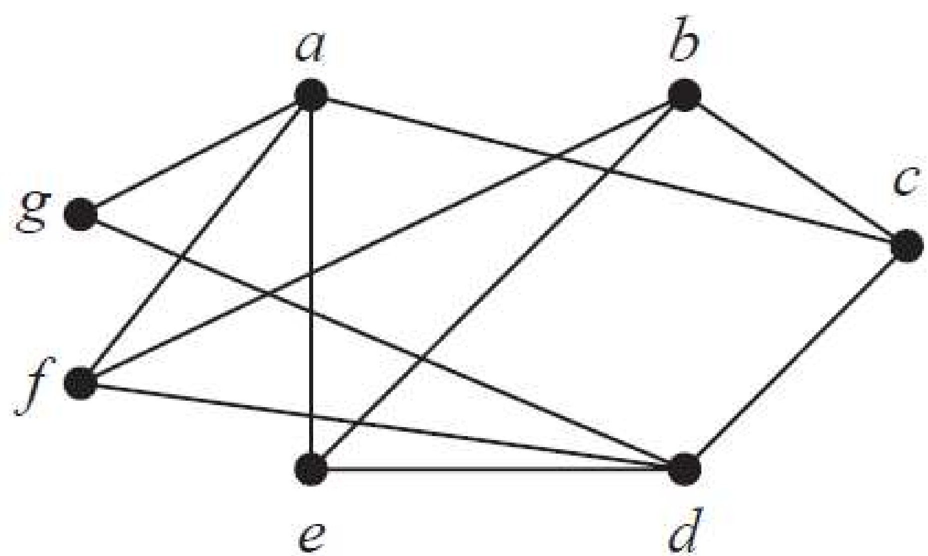
Bipartite Graphs

- **Whether this is a bipartite graph ?**



Bipartite Graphs

- **Whether this is a bipartite graph ?**



Graph G is bipartite because its vertex set is the union of two disjoint sets, {a, b, d} and {c, e, f, g}

THEOREM

- A **simple graph is bipartite** if and only if it is possible to assign **one of two different colors to each vertex of the graph** so that **no two adjacent vertices are assigned the same color**.
- ***Proof:***

THEOREM

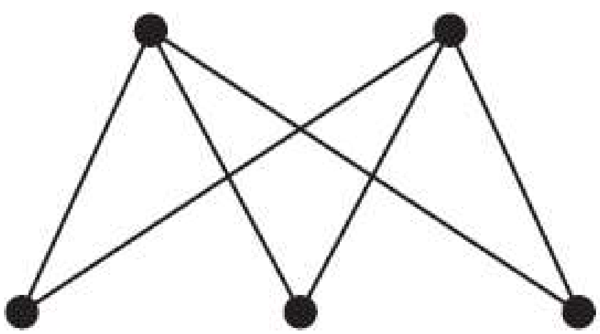
- First, **suppose that G = (V ,E) is a bipartite simple graph**.
- Then **V = V1** ∪ **V2**, where **V1 and V2 are disjoint sets and every edge in E connects a vertex in V1 and a vertex in V2**.
- If we assign **one color to each vertex in V1** and a **second color to each vertex in V2**, then **no two adjacent vertices are assigned the same color**.

THEOREM

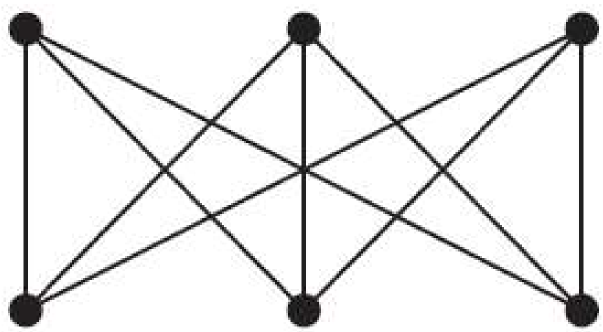
- Now **suppose that it is possible to assign colors to the vertices of the graph using just two colors** so that **no two adjacent vertices are assigned the same color**.
- Let **V1 be the set of vertices assigned one color** and **V2 be the set of vertices assigned the other color**. Then, **V1 and V2 are disjoint** and **V = V1 U V2**.
- Furthermore, **every edge connects** a vertex in **V1** and a vertex in **V2** because **no two adjacent vertices are either both in V1 or both in V2**.
- Consequently, **G is bipartite**.

Complete Bipartite Graphs

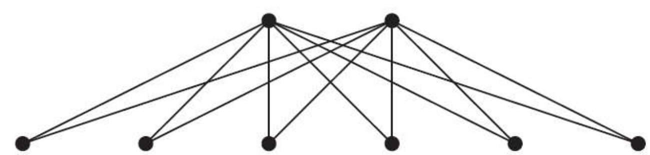
- A **complete bipartite graph $K_{m,n}$ is a graph** that has its **vertex set partitioned into two subsets of m and n vertices**, respectively with **a vertex in the set m is connected to every vertex in the set n**.



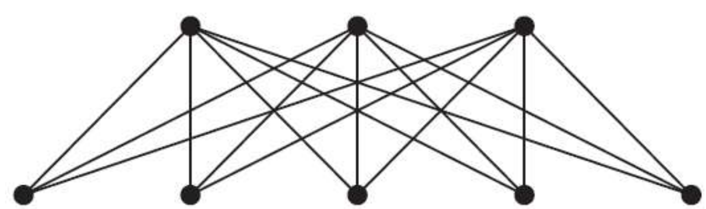
$K_{2,3}$



$K_{3,3}$



$K_{2,6}$



$K_{3,5}$

Bipartite Graphs and Matching

- Bipartite graphs can be used to model many types of applications that involve **matching the elements of one set to elements of another.**

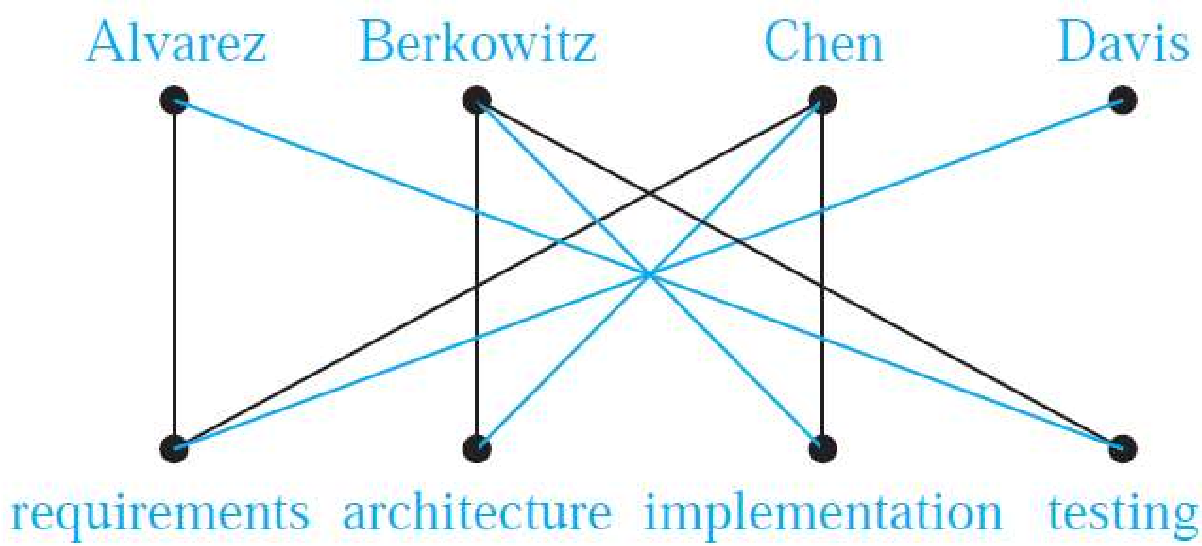
Job Assignments

- 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.
- We represent each **employee** by a **vertex** and each **job** by a **vertex**.
- **Edge** from the **employee to all jobs** that the employee has been **trained to do**.
- We can have **two disjoint sets**, the **set of employees** and the **set of jobs**.
- 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.

Job Assignments

- Suppose that a group has four employees: **Alvarez, Berkowitz, Chen, and Davis**;
- Suppose that four jobs need to be done to complete Project 1:

Requirements, architecture, implementation, and testing.

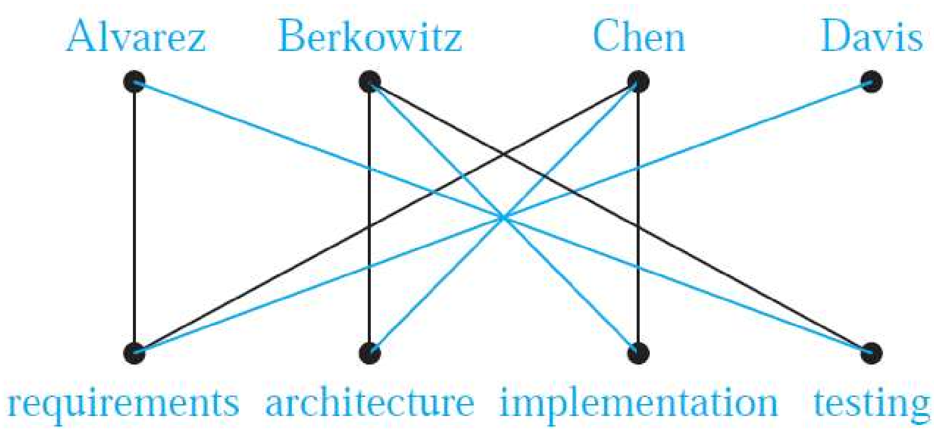


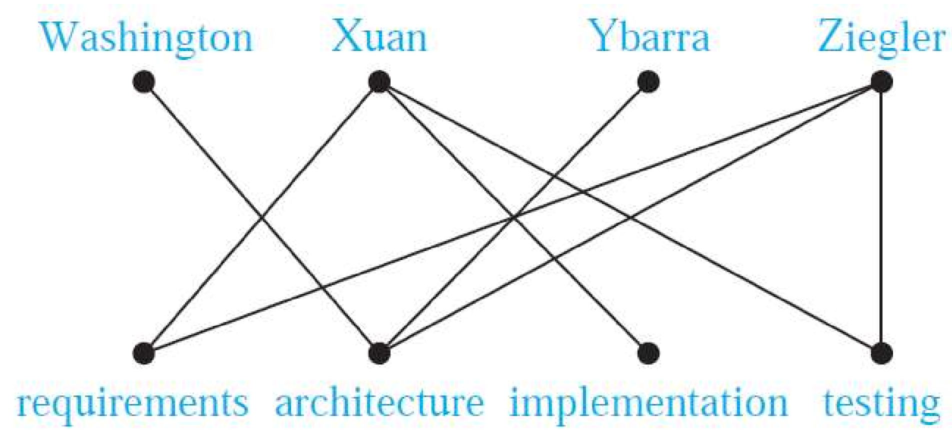
Matching

- **Finding an assignment of jobs** to employees can be thought of as **finding a matching in the graph model**, where a **matching M** in a simple graph G = (V ,E) is a **subset of the set E of edges of the graph** such that **no two edges are incident with the same vertex**.
- In other words, a **matching is a subset of edges** such that **if {s, t} and {u, v} are distinct edges** of the matching, then **s, t , u, and v are distinct**.
- A vertex that is the **endpoint of an edge of a matching M** is said to be **matched in M**; otherwise it is said to be **unmatched**.

Matching

- A **maximum matching** is a matching **with the largest number of edges**.
- We say that a matching M in a bipartite graph G = (V ,E) with bipartition (V1, V2) is a **complete matching** from V1 to V2 **if every vertex in V1 is the endpoint of an edge in the matching, or equivalently**, **if |M| = |V1|**.



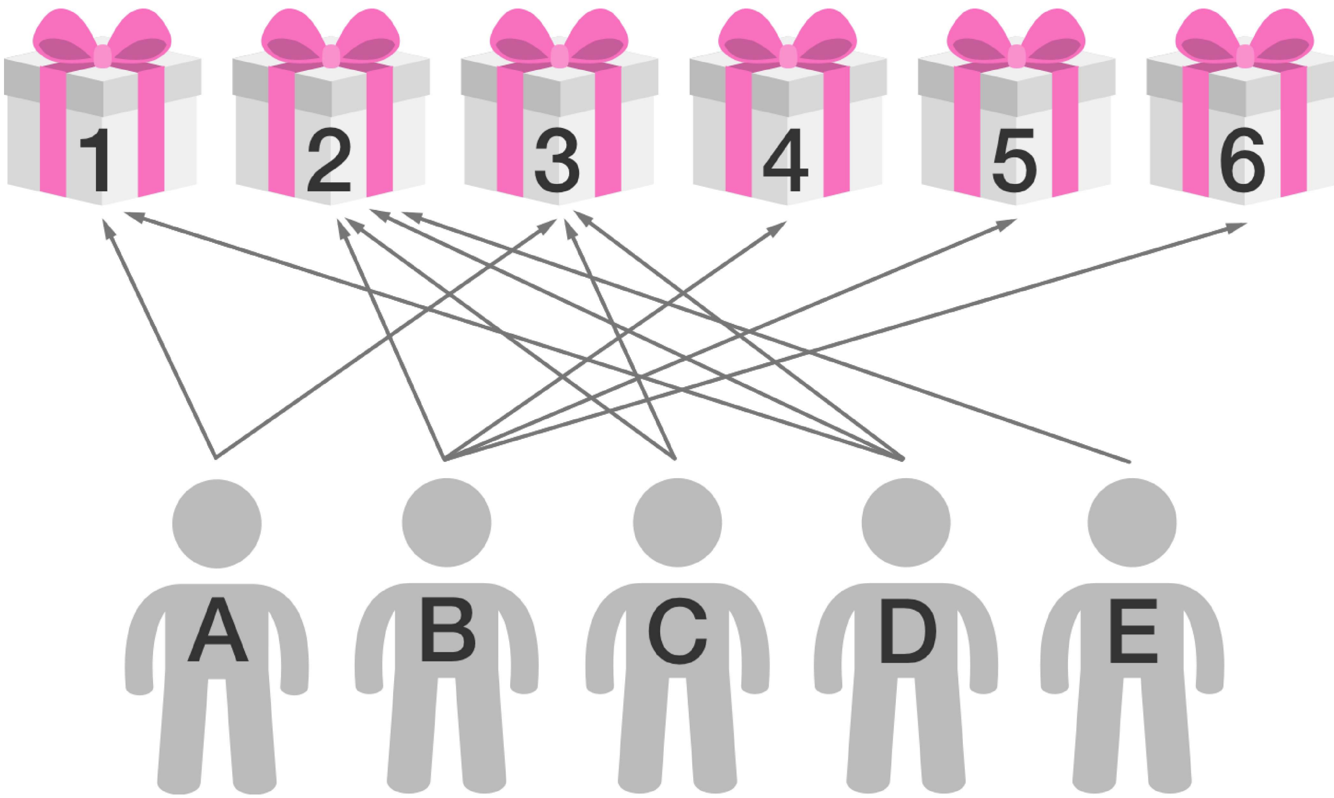


Marriages on an Island-Matching

- **Suppose that there are m men and n women on an island.**
- Each person has a list of members of the opposite gender **acceptable as a spouse**.
- We construct a bipartite graph G = (V1, V2) where **V1 is the set of men** and **V2 is the set of women** so that there is an **edge between a man and a woman if they find each other acceptable as a spouse**.
- A **matching in this graph consists** of a set of edges, where each pair of endpoints of an edge is a **husband-wife pair**.
- A **maximum matching** is a **largest possible set of married couples**, and a **complete matching** **of V1** is a **set of married couples where every man is married**, but **possibly not all women**.

HALL'S MARRIAGE THEOREM

- The **bipartite graph G = (V ,E)** with bipartition (V1, V2) has a complete matching from **V1 to V2** if and only if **|N(A)| ≥ |A|** **for all subsets A of V1**.



{A}->{1,3}
{A,B}->{1,2,3,4,5,6}
{A,B,C,D,E}->{1,2,3,4,5,6}
{A,C,D,E}->{1,2,3}

HALL'S MARRIAGE THEOREM

- The **bipartite graph G = (V ,E)** with bipartition (V1, V2) has a complete matching from **V1 to V2** if and only if **|N(A)| ≥ |A|** **for all subsets A of V1**.

Proof

- We first prove the **only if part of the theorem**. To do so, suppose that **there is a complete matching M from V1 to V2.** Then, if **A ⊆ V1**, for every vertex **v ∈ A**, there is an edge in M connecting v to a vertex in V2. Consequently, **there are at least as many vertices in V2 that are neighbors of vertices in V1** as there are vertices in V1. It follows that **|N(A)| ≥ |A|.**

HALL'S MARRIAGE THEOREM

- The **bipartite graph G = (V ,E)** with bipartition (V1, V2) has a complete matching from **V1 to V2** if and only if **|N(A)| ≥ |A|** **for all subsets A of V1**.

Proof

- To **prove the if part** of the theorem, the more difficult part, we need to show that **if |N(A)| ≥ |A| for all A ⊆ V1**, then **there is a complete matching M from V1 to V2**. We will use strong induction on |V1| to prove this.

HALL'S MARRIAGE THEOREM

Basis step:

- If **$|V1| = 1$**, then V1 contains a single vertex v_0. Because $|N(\{v_0\})| \geq |\{v_0\}| = 1$, there is at least one edge connecting v_0 and a vertex $w_0 \in$ V2. Any such edge forms a complete matching from V1 to V2.

Inductive step :

We first state the inductive hypothesis.

Inductive hypothesis:

Let **k be a positive integer**. If G = (V ,E) is a bipartite graph with bipartition (V1, V2), and **$|V1| = j \leq k$**, then **there is a complete matching M from V1 to V2** whenever the condition that **$|N(A)| \geq |A|$ for all $A \subseteq V1$ is met**.

HALL'S MARRIAGE THEOREM

- Now suppose that **H = (W, F) is a bipartite graph** with **bipartition (W1,W2)** and **|W1| = k + 1**. We will prove that the inductive holds using a proof by cases, using two case.
- **Case (i)** applies when for **all integers j with 1 ≤ j ≤ k**, the **vertices in every set of j elements from W1** are **adjacent to at least j + 1 elements of W2**.
- **Case (ii)** applies when for **some j with 1 ≤ j ≤ k** there is a subset W1 of j vertices such that there are **exactly j neighbors of these vertices in W2**.
- Because **either Case (i) or Case (ii) holds**, we need only consider these cases to complete the inductive step.

HALL'S MARRIAGE THEOREM

Case (i):

- Suppose that for **all integers j with 1 ≤ j ≤ k**, the vertices in every **subset of j elements from W1 are adjacent to at least j + 1 elements of W2.**
- Then, we **select a vertex v ∈ W1** and **an element w ∈ N({v})**, which must exist by our assumption that **|N({v}| ≥ |{v}| = 1.**
- We **delete v and w and all edges incident to them from H**. This produces a bipartite graph H with bipartition **(W1 − {v},W2 − {w})**. Because **|W1 − {v}| = k**, the inductive hypothesis tells **us there is a complete matching from W1 − {v} to W2 − {w}.** Adding the edge from v to w to this complete matching produces a complete matching from W1 to W2.

HALL'S MARRIAGE THEOREM

Case (ii):

Suppose that for **some j with $1 \le j \le k$**, **there is a subset W11 of j vertices such that there are exactly j neighbors of these vertices in W2**.

Let **W22 be the set of these neighbors**. Then, **by the inductive hypothesis there is a complete matching from W1 to W2. Remove these 2j vertices from W1 and W2 and all incident edges to produce a bipartite graph K** with bipartition **(W1 − W11, W2 − W22)**.

- **Graph K satisfies the condition $|N(A)| \ge |A|$ for all subsets A of W1 − W11**. **If not**, there would be a subset of t vertices of W1 − W11 where $1 \le t \le k + 1 - j$ such that the vertices in this subset have fewer than t vertices of W2 − W22 as neighbors. **contradicting the hypothesis that $|N(A)| \ge |A|$ for all $A \subseteq$ W1**.

Stable marriage problem

- The **stable marriage problem** (also **stable matching problem** or **SMP**) is the **problem of finding a stable matching between two equally sized sets of elements given an ordering of preferences for each element**.
- A **matching** is a **mapping from the elements of one set to the elements of the other set**.
- A matching is ***not* stable if**:
  - There is an element *A* of the first matched set which prefers some given element *B* of the second matched set over the element to which *A* is already matched, and
  - *B* also prefers *A* over the element to which *B* is already matched.

Stable marriage problem

- Let there be two men m1 and m2 and two women w1 and w2.
  Let m1's list of preferences be {w1, w2}
  Let m2's list of preferences be {w1, w2}
  Let w1's list of preferences be {m1, m2}
  Let w2's list of preferences be {m1, m2}
  - The matching { {m1, w2}, {w1, m2} } **is not stable** because m1 and w1 would prefer each other over their assigned partners.

Stable marriage problem

- **Gale–Shapley algorithm**
- In the first round, first

a) each unengaged man proposes to the woman he prefers most, and then

b) each woman replies "maybe" to her suitor she most prefers and "no" to all other suitors. She is then provisionally "engaged" to the suitor she most prefers so far, and that suitor is likewise provisionally engaged to her.

Stable marriage problem

- **Gale–Shapley algorithm**
- In each subsequent round, first

a) each unengaged man proposes to the most-preferred woman to whom he has not yet proposed (regardless of whether the woman is already engaged), and then

b) each woman replies "maybe" if she is currently not engaged or if she prefers this man over her current provisional partner (in this case, she rejects her current provisional partner who becomes unengaged). The provisional nature of engagements preserves the right of an already-engaged woman to "trade up" (and, in the process, to "jilt" her until-then partner).

- This process is repeated until everyone is engaged.

Gale–Shapley algorithm

The runtime complexity of this algorithm is n^2 where n is the number of men or women.

Initialize all men and women to free

while there exist a free man m who still has a woman w to propose to

{

w = m's highest ranked such woman to whom he has not yet proposed

if w is free

    (m, w) become engaged

else some pair (m', w) already exists

    if w prefers m to m'

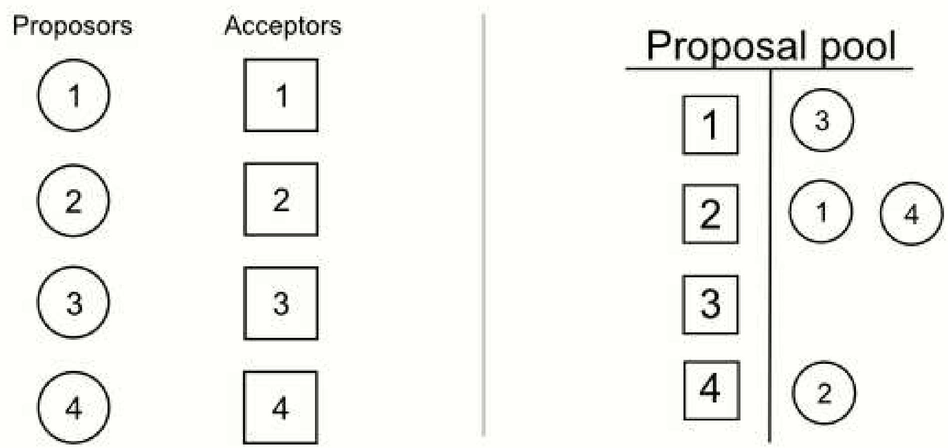
        (m, w) become engaged

        m' becomes free

    else (m', w) remain engaged }

Gale–Shapley algorithm

Round : 1



- 1-4 propose, as none are currently tentatively attached

Preferences

□ → ○

Acceptor Table

1	1	3	2	4
2	3	4	1	2
3	4	2	3	1
4	3	2	1	4

○ → □

Proposor Table

1	2	1	3	4
2	4	1	2	3
3	1	3	2	4
4	2	3	1	4

Reference

- **Rosen, Kenneth H., and Kamala Krithivasan. Discrete mathematics and its applications: with combinatorics and graph theory. Tata McGraw-Hill Education, 2016.**