Discrete Optimization
Graphs

Ngày 1 tháng 8 năm 2011

Lecture 4: Graphs

Graphs are important structures with numerous applications in many areas. We shall be using them extensively in this class.

Lecture 4: Graphs

Graphs are important structures with numerous applications in many areas. We shall be using them extensively in this class. In this lecture we shall review standard definitions, examples, prove some basic theorems and prepare to study some fundamental algorithms.

Lecture 4: Graphs

Graphs are important structures with numerous applications in many areas. We shall be using them extensively in this class. In this lecture we shall review standard definitions, examples, prove some basic theorems and prepare to study some fundamental algorithms.

Definition

A finite **graph** *is a pair* (V, E) *where* V *is a finite set called* **vertices**, E *is a family of pairs* $(x, y) \mid x, y \in V$ *called* **edges**.

Lecture 4: Graphs

Graphs are important structures with numerous applications in many areas. We shall be using them extensively in this class. In this lecture we shall review standard definitions, examples, prove some basic theorems and prepare to study some fundamental algorithms.

Definition

A finite **graph** *is a pair (V, E) where V is a finite set called* **vertices**, *E is a family of pairs $(x, y) \mid x, y \in V$ called* **edges**.

Lecture 4: Graphs

Graphs are important structures with numerous applications in many areas. We shall be using them extensively in this class. In this lecture we shall review standard definitions, examples, prove some basic theorems and prepare to study some fundamental algorithms.

Definition

*A finite **graph** is a pair (V, E) where V is a finite set called **vertices**, E is a family of pairs $(x, y) \mid x, y \in V$ called **edges**.*

We shall usually denote graphs by $G(V, E)$ or $H(V, E)$ etc.

Remark

We used the word family rather than set to account for the possibility that a pair (x,y) will apear more than once. Also we used (x,y) rather than $\{x, y\}$ to account for a pair (x,x).

Examples

Graphs are dynamic data structures. They can grow (adding vertice and edges) or shrink (deleting edges or vertices). When a vertex is delted, all edges incident with it are also deleted.

Examples

Graphs are dynamic data structures. They can grow (adding vertice and edges) or shrink (deleting edges or vertices). When a vertex is delted, all edges incident with it are also deleted.

- $V = \{All\ students\ at\ HUS\}$
  $E = \{(x, y) |\ x\ and\ y\ attend\ the\ same\ class\}$

Examples

Graphs are dynamic data structures. They can grow (adding vertice and edges) or shrink (deleting edges or vertices). When a vertex is delted, all edges incident with it are also deleted.

- $V = \{\textit{All students at HUS}\}$
  $E = \{(x, y) |\ x \textit{ and } y \textit{ attend the same class}\}$
- $V = \{\textit{All cities in Vietnam}\}$
  $E = \{(x, y) |\ \textit{There is a direct flight from } x \textit{ to } y\}$.

Examples

Graphs are dynamic data structures. They can grow (adding vertice and edges) or shrink (deleting edges or vertices). When a vertex is delted, all edges incident with it are also deleted.

- $V = \{\textit{All students at HUS}\}$
  $E = \{(x, y) \mid x \textit{ and } y \textit{ attend the same class}\}$
- $V = \{\textit{All cities in Vietnam}\}$
  $E = \{(x, y) \mid \textit{ There is a direct flight from } x \textit{ to } y\}$.
- $V = N \cup D$ *where:* $N = \{\textit{All nurses in a hospital}\}$
  $D = \{\textit{all nurses' duties}\}$
  $E = \{(x, y) \mid x \textit{ is a nurse, }$ *y is a duty which x is qualified to perform*$\}$.

Examples

Graphs are dynamic data structures. They can grow (adding vertice and edges) or shrink (deleting edges or vertices). When a vertex is delted, all edges incident with it are also deleted.

- $V = \{\textit{All students at HUS}\}$
  $E = \{(x, y) |\ x \textit{ and } y \textit{ attend the same class}\}$
- $V = \{\textit{All cities in Vietnam}\}$
  $E = \{(x, y) |\ \textit{There is a direct flight from } x \textit{ to } y\}.$
- $V = N \cup D$ *where:* $N = \{\textit{All nurses in a hospital}\}$
  $D = \{\textit{all nurses' duties}\}$
  $E = \{(x, y) | x \textit{ is a nurse, } \ y \text{ is a duty which } x \text{ is qualified to perform}\}.$
- $V = \{\{x, y\} | \{x, y\} \subset \{1, 2, \ldots k\}\}$
  $E = \{(\{x, y\}, \{u, w\}) | \{x, y\} \cap \{u, w\} = \emptyset.\}$

Examples

Graphs are dynamic data structures. They can grow (adding vertice and edges) or shrink (deleting edges or vertices). When a vertex is delted, all edges incident with it are also deleted.

- $V = \{\textit{All students at HUS}\}$
  $E = \{(x, y) |\ x \textit{ and } y \textit{ attend the same class}\}$
- $V = \{\textit{All cities in Vietnam}\}$
  $E = \{(x, y) |\ \textit{There is a direct flight from } x \textit{ to } y\}.$
- $V = N \cup D$ *where:* $N = \{\textit{All nurses in a hospital}\}$
  $D = \{\textit{all nurses' duties}\}$
  $E = \{(x, y) | x \textit{ is a nurse, } \ y \text{ is a duty which } x \text{ is qualified to perform}\}.$
- $V = \{\{x, y\} | \{x, y\} \subset \{1, 2, \dots k\}\}$
  $E = \{(\{x, y\}, \{u, w\}) | \{x, y\} \cap \{u, w\} = \emptyset.\}$
- $V = GF^*(q) \qquad E = \{(x, y) | x - y \in QR(q)\}.$

Graph Basics

Definition

Graph Basics

Definition

- *B-1: We use the term* **graph** *for $G(V, E)$ if the edges (x, y) are unordered pairs.*

Graph Basics

Definition

- *B-1: We use the term* **graph** *for* $G(V, E)$ *if the edges* (x, y) *are unordered pairs.*
- *The students, nurses, number pairs and quadratic residues for* $q \equiv 1 \mod 4$ *examples are simple graphs.*

Graph Basics

Definition

- *B-1: We use the term* **graph** *for $G(V, E)$ if the edges (x, y) are unordered pairs.*
- *The students, nurses, number pairs and quadratic residues for $q \equiv 1 \mod 4$ examples are simple graphs.*
- *B-2: If the edges of $G(V, E)$ are ordered pairs (oriented) we call $G(V, E)$ a* **digraph** *(directed graph).*

Graph Basics

Definition

- *B-1: We use the term* **graph** *for $G(V, E)$ if the edges (x, y) are unordered pairs.*
- *The students, nurses, number pairs and quadratic residues for $q \equiv 1 \bmod 4$ examples are simple graphs.*
- *B-2: If the edges of $G(V, E)$ are ordered pairs (oriented) we call $G(V, E)$ a* **digraph** *(directed graph).*
- *The cities graph is a directed multi-graph and the quadratic residues for $q \equiv 3 \bmod 4$ is a digraph.*

Graph Basics

Definition

- *B-1: We use the term* **graph** *for $G(V, E)$ if the edges (x, y) are unordered pairs.*
- *The students, nurses, number pairs and quadratic residues for $q \equiv 1 \mod 4$ examples are simple graphs.*
- *B-2: If the edges of $G(V, E)$ are ordered pairs (oriented) we call $G(V, E)$ a* **digraph** *(directed graph).*
- *The cities graph is a directed multi-graph and the quadratic residues for $q \equiv 3 \mod 4$ is a digraph.*
- *B-3: The degree of a vertex $v \in V(G)$, denoted by $d_G(v)$ is the number of vertices adjacent to v in G $(d_G(v) = |\{x|(v, x) \in E(G)\}|)$.*

Graph Basics

Definition

- *B-1: We use the term* **graph** *for $G(V, E)$ if the edges (x, y) are unordered pairs.*
- *The students, nurses, number pairs and quadratic residues for $q \equiv 1 \mod 4$ examples are simple graphs.*
- *B-2: If the edges of $G(V, E)$ are ordered pairs (oriented) we call $G(V, E)$ a* **digraph** *(directed graph).*
- *The cities graph is a directed multi-graph and the quadratic residues for $q \equiv 3 \mod 4$ is a digraph.*
- *B-3: The degree of a vertex $v \in V(G)$, denoted by $d_G(v)$ is the number of vertices adjacent to v in G ($d_G(v) = |\{x|(v, x) \in E(G)\}|$).*
- *B-4: a graph G is* **r-regular** *if all vertices have degree r.*

Graph Basics

Definition

- *B-1: We use the term* **graph** *for $G(V,E)$ if the edges (x,y) are unordered pairs.*
- *The students, nurses, number pairs and quadratic residues for $q \equiv 1 \mod 4$ examples are simple graphs.*
- *B-2: If the edges of $G(V,E)$ are ordered pairs (oriented) we call $G(V,E)$ a* **digraph** *(directed graph).*
- *The cities graph is a directed multi-graph and the quadratic residues for $q \equiv 3 \mod 4$ is a digraph.*
- *B-3: The degree of a vertex $v \in V(G)$, denoted by $d_G(v)$ is the number of vertices adjacent to v in G ($d_G(v) = |\{x|(v,x) \in E(G)\}|$).*
- *B-4: a graph G is* **r-regular** *if all vertices have degree r.*
- *B-5: A graph G is* **labeled** *if the vertices are assigned unique names.*

Paths, walks

Definition

Paths, walks

Definition

- *G-1: An edge of the form* (x, x) *is called a* **loop**.

Paths, walks

Definition

- *G-1: An edge of the form* (x, x) *is called a* **loop**.
- *G-2: If a graph contains the edge* (x, y) *more than once, we say that it has* **parallel edges**.

Paths, walks

Definition

- *G-1: An edge of the form* (x, x) *is called a* **loop**.
- *G-2: If a graph contains the edge* (x, y) *more than once, we say that it has* **parallel edges**.
- *G-3: A loop-less graph without parallel edges is a* **simple graph**.

Paths, walks

Definition

- *G-1: An edge of the form (x, x) is called a* **loop**.
- *G-2: If a graph contains the edge (x, y) more than once, we say that it has* **parallel edges**.
- *G-3: A loop-less graph without parallel edges is a* **simple graph**.
- *G-4: A* **walk** *in a graph is a sequence $(v_0, E_1, v_1, E_2, \ldots, v_{k-1}, E_k, v_k)$ such that $E_i = (v_{i-1}, v_i)$. The* **length** *of a walk is the number of edges it contains.*

Paths, walks

Definition

- *G-1: An edge of the form (x, x) is called a* **loop**.
- *G-2: If a graph contains the edge (x, y) more than once, we say that it has* **parallel edges**.
- *G-3: A loop-less graph without parallel edges is a* **simple graph**.
- *G-4: A* **walk** *in a graph is a sequence $(v_0, E_1, v_1, E_2, \ldots, v_{k-1}, E_k, v_k)$ such that $E_i = (v_{i-1}, v_i)$. The* **length** *of a walk is the number of edges it contains.*
- *G-5: A walk in $G(V, E)$ in which all vertices are distinct is a* **path**.

Paths, walks

Definition

- *G-1: An edge of the form (x, x) is called a* **loop**.
- *G-2: If a graph contains the edge (x, y) more than once, we say that it has* **parallel edges**.
- *G-3: A loop-less graph without parallel edges is a* **simple graph**.
- *G-4: A* **walk** *in a graph is a sequence $(v_0, E_1, v_1, E_2, \dots, v_{k-1}, E_k, v_k)$ such that $E_i = (v_{i-1}, v_i)$. The* **length** *of a walk is the number of edges it contains.*
- *G-5: A walk in $G(V, E)$ in which all vertices are distinct is a* **path**.
- *G-6: A walk $(v_0, E_1, v_1, E_2, \dots, v_{k-1}, E_k, v_k)$ in which $v_1 = v_k$ is a* **closed walk**.

Paths, walks

Definition

- *G-1: An edge of the form (x, x) is called a* **loop**.
- *G-2: If a graph contains the edge (x, y) more than once, we say that it has* **parallel edges**.
- *G-3: A loop-less graph without parallel edges is a* **simple graph**.
- *G-4: A* **walk** *in a graph is a sequence $(v_0, E_1, v_1, E_2, \ldots, v_{k-1}, E_k, v_k)$ such that $E_i = (v_{i-1}, v_i)$. The* **length** *of a walk is the number of edges it contains.*
- *G-5: A walk in $G(V, E)$ in which all vertices are distinct is a* **path**.
- *G-6: A walk $(v_0, E_1, v_1, E_2, \ldots, v_{k-1}, E_k, v_k)$ in which $v_1 = v_k$ is a* **closed walk**.
- *G-7: A closed walk in which $v_i \neq v_j, \mathit{for}\ i \neq j,\ 0 \leq i, j\ < k$ is a* **cycle**.

Digraphs Basics

- D-1: In a digraph D $in_D(v) = |\{x|(x, v) \in E(D)\}|$ is the indegree of v and $out_D(v) = |\{x|(v, x) \in E(D)\}|$ is the outdegree.

Digraphs Basics

- D-1: In a digraph D $in_D(v) = |\{x|(x,v) \in E(D)\}|$ is the indegree of v and $out_D(v) = |\{x|(v,x) \in E(D)\}|$ is the outdegree.
- A digraph D is **strongly connected** if between any two vertices x, y there are directed paths $x \to x_1 \to \ldots \to y$ and $y \to y_1 \to \ldots \to x$

Digraphs Basics

- D-1: In a digraph D $in_D(v) = |\{x|(x,v) \in E(D)\}|$ is the indegree of v and $out_D(v) = |\{x|(v,x) \in E(D)\}|$ is the outdegree.
- A digraph D is **strongly connected** if between any two vertices x, y there are directed paths $x \to x_1 \to \ldots \to y$ and $y \to y_1 \to \ldots \to x$
- A **tournament** is a digraph in which between any two vertices there is exactly one directed edge.

Digraphs Basics

- D-1: In a digraph D $in_D(v) = |\{x|(x,v) \in E(D)\}|$ is the indegree of v and $out_D(v) = |\{x|(v,x) \in E(D)\}|$ is the outdegree.
- A digraph D is **strongly connected** if between any two vertices x, y there are directed paths $x \to x_1 \to \ldots \to y$ and $y \to y_1 \to \ldots \to x$
- A **tournament** is a digraph in which between any two vertices there is exactly one directed edge.

Observation

In a digraph D, $\sum_{i=1}^{n} in_D(v_i) = \sum_{i=1}^{n} out_D(v_i)$.

Digraphs Basics

- D-1: In a digraph D $in_D(v) = |\{x|(x, v) \in E(D)\}|$ is the indegree of v and $out_D(v) = |\{x|(v, x) \in E(D)\}|$ is the outdegree.
- A digraph D is **strongly connected** if between any two vertices x, y there are directed paths $x \to x_1 \to \ldots \to y$ and $y \to y_1 \to \ldots \to x$
- A **tournament** is a digraph in which between any two vertices there is exactly one directed edge.

Observation

In a digraph D, $\sum_{i=1}^{n} in_D(v_i) = \sum_{i=1}^{n} out_D(v_i)$.

Theorem

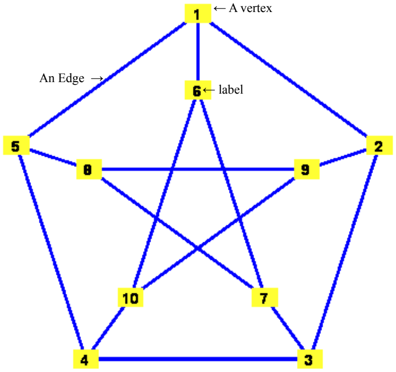
Every tournament has a hamiltonian path.

The Petersen Graph

Petersen's graph.

Petersen's graph has 10 **vertices** and 15 **edges**. This rendition of the graph is **labeled**. Every vertex has 3 **adjacent** vertices (neighbors).

The **degree** of every vertex is 3. This graph is regular of degree 3 (3-regular, or **cubic**).



Using graphs to model problems

Using graphs to model problems

- *1. Hoan is a student in your class. You want to deliver a book to Trung. Hoan and Trung are students at HUS, as such they are vertices in the HUS graph. Is there a path between Hoan and Trung?*

Using graphs to model problems

- *1. Hoan is a student in your class. You want to deliver a book to Trung. Hoan and Trung are students at HUS, as such they are vertices in the HUS graph. Is there a path between Hoan and Trung?*
- *2. Ralph is a tourist visiting Vietnam. He has a list of cities he would like to visit. He would like to start in Hanoi and return to Hanoi. Does the graph contain a cycle through these cities?*

Using graphs to model problems

- *1. Hoan is a student in your class. You want to deliver a book to Trung. Hoan and Trung are students at HUS, as such they are vertices in the HUS graph. Is there a path between Hoan and Trung?*
- *2. Ralph is a tourist visiting Vietnam. He has a list of cities he would like to visit. He would like to start in Hanoi and return to Hanoi. Does the graph contain a cycle through these cities?*
- *3. A group of nurses show up for a shift. Can you match the nurses with duties needed to be performed? Can you find a set of disjoint edges in the nurses graph?*

Using graphs to model problems

- *1. Hoan is a student in your class. You want to deliver a book to Trung. Hoan and Trung are students at HUS, as such they are vertices in the HUS graph. Is there a path between Hoan and Trung?*
- *2. Ralph is a tourist visiting Vietnam. He has a list of cities he would like to visit. He would like to start in Hanoi and return to Hanoi. Does the graph contain a cycle through these cities?*
- *3. A group of nurses show up for a shift. Can you match the nurses with duties needed to be performed? Can you find a set of disjoint edges in the nurses graph?*
- *4. Can a tourist visit all airpots in Vietnam by flying in and out of airports? Is there a closed walk through all vertices?*

Connectivity, Cycles

Definition

Connectivity, Cycles

Definition

- *G-8: A graph $G = G(V, E)$ is* **connected** *if between any pair of vertices there is a path.*

Connectivity, Cycles

Definition

- *G-8: A graph $G = G(V, E)$ is* **connected** *if between any pair of vertices there is a path.*
- *G-9: A path containing all vertices is a* **Hamiltonian path***. If G has such a path we say that G is* **traceable***.*

Connectivity, Cycles

Definition

- *G-8: A graph $G = G(V, E)$ is* **connected** *if between any pair of vertices there is a path.*
- *G-9: A path containing all vertices is a* **Hamiltonian path**. *If G has such a path we say that G is* **traceable**.
- *G-10: G is* **Hamiltonian** *if it has a cycle that contains (visits) every vertex,*

Connectivity, Cycles

Definition

- *G-8: A graph $G = G(V, E)$ is* **connected** *if between any pair of vertices there is a path.*
- *G-9: A path containing all vertices is a* **Hamiltonian path**. *If G has such a path we say that G is* **traceable**.
- *G-10: G is* **Hamiltonian** *if it has a cycle that contains (visits) every vertex,*
- *G-11: G is* **Eulerian** *if it contains a closed walk that uses every edge once.*

Connectivity, Cycles

Definition

- *G-8: A graph $G = G(V, E)$ is* **connected** *if between any pair of vertices there is a path.*
- *G-9: A path containing all vertices is a* **Hamiltonian path***. If G has such a path we say that G is* **traceable***.*
- *G-10: G is* **Hamiltonian** *if it has a cycle that contains (visits) every vertex,*
- *G-11: G is* **Eulerian** *if it contains a closed walk that uses every edge once.*
- *G-12: The* **distance** *between two vertices in a graph is the length of the shortest path between them.*

Connectivity, Cycles

Definition

- *G-8: A graph $G = G(V, E)$ is* **connected** *if between any pair of vertices there is a path.*
- *G-9: A path containing all vertices is a* **Hamiltonian path**. *If G has such a path we say that G is* **traceable**.
- *G-10: G is* **Hamiltonian** *if it has a cycle that contains (visits) every vertex,*
- *G-11: G is* **Eulerian** *if it contains a closed walk that uses every edge once.*
- *G-12: The* **distance** *between two vertices in a graph is the length of the shortest path between them.*
- *G-13: The* **diameter** *of a connected graph is the length of the longest distance among all pairs of vertices.*

Connectivity

Connectivity

- *The relation "there is a walk between two vertices x and y" is an equivalence relation on $V(G)$.*

Connectivity

- *The relation “there is a walk between two vertices x and y” is an equivalence relation on $V(G)$.*

Connectivity

- *The relation "there is a walk between two vertices x and y" is an equivalence relation on $V(G)$.*

Definition

Connectivity

- *The relation "there is a walk between two vertices x and y" is an equivalence relation on $V(G)$.*

Definition

- *The equivalence classes induced by this relation are the* **connected components** *of the graph $G(V, E)$.*

Connectivity

- *The relation "there is a walk between two vertices x and y" is an equivalence relation on $V(G)$.*

Definition

- - *The equivalence classes induced by this relation are the* **connected components** *of the graph $G(V, E)$.*
    In a connected graph G a vertex $v \in G(V, E)$ whose removal disconnects G is called a **cut vertex** *(or an articulation point).*

Connectivity

- *The relation "there is a walk between two vertices x and y" is an equivalence relation on $V(G)$.*

Definition

- - *The equivalence classes induced by this relation are the* **connected components** *of the graph $G(V,E)$.*
    In a connected graph G a vertex $v \in G(V,E)$ whose removal disconnects G is called a **cut vertex** *(or an articulation point).*

Definition

A graph G is **r-connected** *if $G = K_r$ or G cannot be disconnected by removing less than r vertices.*

Cycles, Notation

Definition (Notation)

Cycles, Notation

Definition (Notation)

- *a cycle of lnegth k is denoted by C_k.*

Cycles, Notation

Definition (Notation)

- *a cycle of lnegth k is denoted by C_k.*
- *C_3 is a triangle, C_4 a quadrilateral, C_5 a pentagon etc.*

Cycles, Notation

Definition (Notation)

- *a cycle of lnegth k is denoted by C_k.*
- *C_3 is a triangle, C_4 a quadrilateral, C_5 a pentagon etc.*
- *A graph $G(V, E)$ has* **girth** *k if the shortest cycle in G has length k.*

Cycles, Notation

Definition (Notation)

- *a cycle of lnegth k is denoted by C_k.*
- *C_3 is a triangle, C_4 a quadrilateral, C_5 a pentagon etc.*
- *A graph $G(V, E)$ has* **girth** *k if the shortest cycle in G has length k.*
- *The complete graph of order n is denoted by by K_n.*

Cycles, Notation

Definition (Notation)

- *a cycle of lnegth k is denoted by C_k.*
- *C_3 is a triangle, C_4 a quadrilateral, C_5 a pentagon etc.*
- *A graph $G(V, E)$ has* **girth** *k if the shortest cycle in G has length k.*
- *The complete graph of order n is denoted by by K_n.*
- *The n-wheel W_n is a graph consisting of C_n and one additional vertex connected to all vertices of the cycle C_n.*

Cycles, Notation

Definition (Notation)

- *a cycle of lnegth k is denoted by C_k.*
- *C_3 is a triangle, C_4 a quadrilateral, C_5 a pentagon etc.*
- *A graph $G(V, E)$ has* **girth** *k if the shortest cycle in G has length k.*
- *The complete graph of order n is denoted by by K_n.*
- *The n-wheel W_n is a graph consisting of C_n and one additional vertex connected to all vertices of the cycle C_n.*
- *A set of vertex disjoint edges is a* **matching***.*

Cycles, Notation

Definition (Notation)

- *a cycle of lnegth k is denoted by C_k.*
- *C_3 is a triangle, C_4 a quadrilateral, C_5 a pentagon etc.*
- *A graph $G(V, E)$ has* **girth** *k if the shortest cycle in G has length k.*
- *The complete graph of order n is denoted by by K_n.*
- *The n-wheel W_n is a graph consisting of C_n and one additional vertex connected to all vertices of the cycle C_n.*
- *A set of vertex disjoint edges is a* **matching**.
- *A matching M in G is* **perfect** *if every vertex is incident with one of the edges from M.*

Cycles, Notation

Definition (Notation)

- *a cycle of lnegth k is denoted by C_k.*
- *C_3 is a triangle, C_4 a quadrilateral, C_5 a pentagon etc.*
- *A graph $G(V, E)$ has* **girth** *k if the shortest cycle in G has length k.*
- *The complete graph of order n is denoted by by K_n.*
- *The n-wheel W_n is a graph consisting of C_n and one additional vertex connected to all vertices of the cycle C_n.*
- *A set of vertex disjoint edges is a* **matching**.
- *A matching M in G is* **perfect** *if every vertex is incident with one of the edges from M.*

Example

Cycles, Notation

Definition (Notation)

- *a cycle of lnegth k is denoted by C_k.*
- *C_3 is a triangle, C_4 a quadrilateral, C_5 a pentagon etc.*
- *A graph $G(V, E)$ has* **girth** *k if the shortest cycle in G has length k.*
- *The complete graph of order n is denoted by by K_n.*
- *The n-wheel W_n is a graph consisting of C_n and one additional vertex connected to all vertices of the cycle C_n.*
- *A set of vertex disjoint edges is a* **matching**.
- *A matching M in G is* **perfect** *if every vertex is incident with one of the edges from M.*

Example

- *The Petersen graph has a perfect matching, girth* 5.

Cycles, Notation

Definition (Notation)

- *a cycle of lnegth k is denoted by C_k.*
- *C_3 is a triangle, C_4 a quadrilateral, C_5 a pentagon etc.*
- *A graph $G(V, E)$ has* **girth** *k if the shortest cycle in G has length k.*
- *The complete graph of order n is denoted by by K_n.*
- *The n-wheel W_n is a graph consisting of C_n and one additional vertex connected to all vertices of the cycle C_n.*
- *A set of vertex disjoint edges is a* **matching***.*
- *A matching M in G is* **perfect** *if every vertex is incident with one of the edges from M.*

Example

- *The Petersen graph has a perfect matching, girth 5.*
- *The 5-prism has a perfect matching and girth 4.*

More Examples

(Using graphs to model problems)

More Examples

(Using graphs to model problems)

- *You wish to connect processors in a multiprocessor computer or computers in a building to form a local area network (LAN).*

More Examples

(Using graphs to model problems)

- *You wish to connect processors in a multiprocessor computer or computers in a building to form a local area network (LAN).*
- *Coonections are expensive and the number of connections is physically limited.*

More Examples

(Using graphs to model problems)

- *You wish to connect processors in a multiprocessor computer or computers in a building to form a local area network (LAN).*
- *Coonections are expensive and the number of connections is physically limited.*
- *When designing the connections, what will you try to achieve?*

More Examples

(Using graphs to model problems)

- *You wish to connect processors in a multiprocessor computer or computers in a building to form a local area network (LAN).*
- *Coonections are expensive and the number of connections is physically limited.*
- *When designing the connections, what will you try to achieve?*
- *1. "High" connectivity.*

More Examples

(Using graphs to model problems)

- *You wish to connect processors in a multiprocessor computer or computers in a building to form a local area network (LAN).*
- *Coonections are expensive and the number of connections is physically limited.*
- *When designing the connections, what will you try to achieve?*
- *1. “High” connectivity.*
- *2. Fast communications among processors (computers).*

More Examples

(Using graphs to model problems)

- *You wish to connect processors in a multiprocessor computer or computers in a building to form a local area network (LAN).*
- *Coonections are expensive and the number of connections is physically limited.*
- *When designing the connections, what will you try to achieve?*
- *1. “High” connectivity.*
- *2. Fast communications among processors (computers).*
- *3. Multiple ways to exchange information among all processors (computers).*

More Examples

(Using graphs to model problems)

- *You wish to connect processors in a multiprocessor computer or computers in a building to form a local area network (LAN).*
- *Coonections are expensive and the number of connections is physically limited.*
- *When designing the connections, what will you try to achieve?*
- *1. "High" connectivity.*
- *2. Fast communications among processors (computers).*
- *3. Multiple ways to exchange information among all processors (computers).*
- *4. Fault tolerance.*

Continued

(The Graph)

Continued

(The Graph)

- *To build a mathematical model for tackling this problem we define a graph $G(V, E)$ where $V(G)$ is the set of all computers (or processors) and $E(G)$ is the pairs of computers we will connect by hard wire.*

Continued

(The Graph)

- *To build a mathematical model for tackling this problem we define a graph $G(V, E)$ where $V(G)$ is the set of all computers (or processors) and $E(G)$ is the pairs of computers we will connect by hard wire.*
- *To achieve the first goal we would like the graph to have many paths between any pair of vertices, preferably vertex disjoint.*

Continued

(The Graph)

- *To build a mathematical model for tackling this problem we define a graph $G(V, E)$ where $V(G)$ is the set of all computers (or processors) and $E(G)$ is the pairs of computers we will connect by hard wire.*
- *To achieve the first goal we would like the graph to have many paths between any pair of vertices, preferably vertex disjoint.*
- *To achieve the second goal we would like the graph to have a small diameter.*

Continued

(The Graph)

- *To build a mathematical model for tackling this problem we define a graph $G(V, E)$ where $V(G)$ is the set of all computers (or processors) and $E(G)$ is the pairs of computers we will connect by hard wire.*
- *To achieve the first goal we would like the graph to have many paths between any pair of vertices, preferably vertex disjoint.*
- *To achieve the second goal we would like the graph to have a small diameter.*
- *For the third goal we would like to have many edge-disjoint Hamiltonian cycles in the graph.*

Example

Let us study the case of connecting 10 *computers where each computer can be connected to three other computers.*

Example

Let us study the case of connecting 10 *computers where each computer can be connected to three other computers.*

- *1. The 5-prism:*

Example

Let us study the case of connecting 10 *computers where each computer can be connected to three other computers.*

- *1. The 5-prism:*
- *Between any pair of vertices there are* 3 *vertex disjoint paths.*

Example

Let us study the case of connecting 10 *computers where each computer can be connected to three other computers.*

- *1. The 5-prism:*
- *Between any pair of vertices there are* 3 *vertex disjoint paths.*
- *It has Hamiltonian cycles.*

Example

Let us study the case of connecting 10 *computers where each computer can be connected to three other computers.*

- *1. The 5-prism:*
- *Between any pair of vertices there are* 3 *vertex disjoint paths.*
- *It has Hamiltonian cycles.*
- *Its diameter is* 3*.*

Example

Let us study the case of connecting 10 *computers where each computer can be connected to three other computers.*

- *1. The 5-prism:*
- *Between any pair of vertices there are* 3 *vertex disjoint paths.*
- *It has Hamiltonian cycles.*
- *Its diameter is* 3.
- *2. The Petersen graph.*

Example

Let us study the case of connecting 10 *computers where each computer can be connected to three other computers.*

- *1. The 5-prism:*
- *Between any pair of vertices there are* 3 *vertex disjoint paths.*
- *It has Hamiltonian cycles.*
- *Its diameter is* 3*.*
- *2. The Petersen graph.*
- *Between any pair of vertices there are* 3 *vertex disjoint paths.*

Example

Let us study the case of connecting 10 *computers where each computer can be connected to three other computers.*

- *1. The 5-prism:*
- *Between any pair of vertices there are* 3 *vertex disjoint paths.*
- *It has Hamiltonian cycles.*
- *Its diameter is* 3*.*
- *2. The Petersen graph.*
- *Between any pair of vertices there are* 3 *vertex disjoint paths.*
- *It is traceable but not Hamoltonian.*

Example

Let us study the case of connecting 10 *computers where each computer can be connected to three other computers.*

- *1. The 5-prism:*
- *Between any pair of vertices there are* 3 *vertex disjoint paths.*
- *It has Hamiltonian cycles.*
- *Its diameter is* 3*.*
- *2. The Petersen graph.*
- *Between any pair of vertices there are* 3 *vertex disjoint paths.*
- *It is traceable but not Hamoltonian.*
- *Its diameter is* 2

Example

Let us study the case of connecting 10 *computers where each computer can be connected to three other computers.*

- *1. The 5-prism:*
- *Between any pair of vertices there are* 3 *vertex disjoint paths.*
- *It has Hamiltonian cycles.*
- *Its diameter is* 3*.*
- *2. The Petersen graph.*
- *Between any pair of vertices there are* 3 *vertex disjoint paths.*
- *It is traceable but not Hamoltonian.*
- *Its diameter is* 2

Example

Let us study the case of connecting 10 *computers where each computer can be connected to three other computers.*

- *1. The 5-prism:*
- *Between any pair of vertices there are* 3 *vertex disjoint paths.*
- *It has Hamiltonian cycles.*
- *Its diameter is* 3*.*
- *2. The Petersen graph.*
- *Between any pair of vertices there are* 3 *vertex disjoint paths.*
- *It is traceable but not Hamoltonian.*
- *Its diameter is* 2

So which design is better?

Example continued

Example continued

- *Suppose you want to merge two networks with* 10 *computers each by adding just one connection for each computer. How would you do it?*

Example continued

- *Suppose you want to merge two networks with* 10 *computers each by adding just one connection for each computer. How would you do it?*
- *Start from scratch, designing a graph with* 20 *vertices regular of degree* 4.

Example continued

- *Suppose you want to merge two networks with* 10 *computers each by adding just one connection for each computer. How would you do it?*
- *Start from scratch, designing a graph with* 20 *vertices regular of degree* 4*.*
- *Preserve the previous design and connect each computer to its “clone" (the* **prism***).*

Example continued

- *Suppose you want to merge two networks with* 10 *computers each by adding just one connection for each computer. How would you do it?*
- *Start from scratch, designing a graph with* 20 *vertices regular of degree* 4*.*
- *Preserve the previous design and connect each computer to its “clone" (the* **prism***).*
- *What will be the best design?*

Example continued

- *Suppose you want to merge two networks with* 10 *computers each by adding just one connection for each computer. How would you do it?*
- *Start from scratch, designing a graph with* 20 *vertices regular of degree* 4*.*
- *Preserve the previous design and connect each computer to its “clone" (the* **prism***).*
- *What will be the best design?*

Answer

To be answered by you in the assignment.