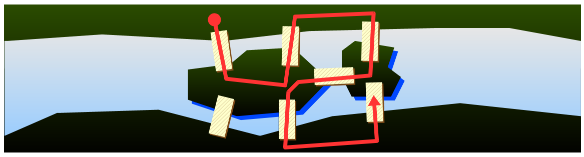
Graph Theory

- **Outline**:
  - **Introduction**
    - Basic terminology and concepts
    - Representations of graphs
    - Operations on Graphs
  - Walks, trails, paths, circuits and cycles
    - Eulerian circuits
    - Hamiltonian cycles
  - Special types of graphs
  - Graph isomorphism and homeomorphism
  - Trees
  - Graph problems and their applications

Slide 1

Graph Theory

- **Introduction and History**
  - Gained sustained interest and dev't only during 1920's.
  - *First paper* on graph theory:
    - published by Leonhard Euler (Swiss) in 1736
    - includes explanation of Königsberg Bridge Problem

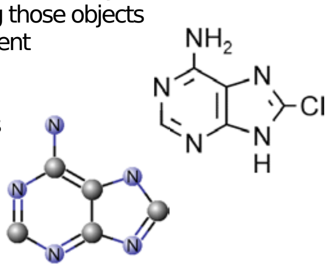


  - First book on graph theory in 1936.

Slide 2

Graph Theory

- **Uses and Applications of Graphs**
  - Graphs are used to represent **objects** and **relationships** among those objects
  - Can be used to represent
    - road systems
    - friendship networks
    - molecular structures
    - soldering lines on circuit boards
    - pattern recognition
    - parallel algorithms



Slide 3

Graphs & Digraphs

- **GRAPH G = {V(G), E(G)}**
  - **G** ≡ **set of vertices** V(G) + **set of edges** E(G)
    - **Order of G** ≡ number of vertices
    - **Size of G** ≡ number of edges
- **Types of graphs**
  - **Directed Graph/Digraph**

    - vertices **u** and **v**
    - *directed edge* **e = (u, v)** [ordered pair]
    - edge (u,v) is **incident from** u
    - edge (u,v) is **incident to** v
    - vertex v is **adjacent to** vertex u

Slide 4

Graphs & Digraphs

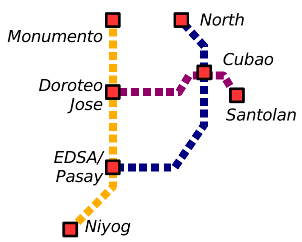
- **GRAPH G = {V(G), E(G)}**
  - **G** ≡ **set of vertices** V(G) + **set of edges** E(G)
    - **Order of G** ≡ number of vertices
    - **Size of G** ≡ number of edges
- **Types of graphs**
  - **<u>Un</u>directed Graph**

    - vertices **u** and **v**
    - *undirected edge* **e = (u, v)** or **(v,u)**
    - edge (u,v) (or (v,u)) is **incident on** both u and v
    - v is **adjacent to** u and u is adjacent to v
    - **adjacent edges** ≡ at least one common vertex

Slide 5

Graphs & Digraphs

- ***Example***: Manila MRT system

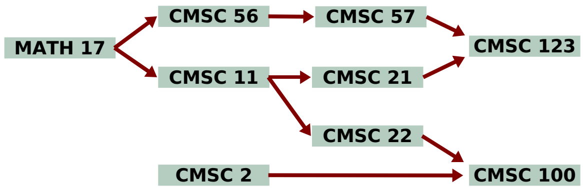


V(G) = {M, N, P, C, O, S, D}
E(G) = {(M, D), (D, P), (D, C), (P, N), (P, C), (C, S), (O, C)}

Slide 6

Graphs & Digraphs

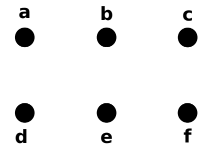
- ***Example*:** Courses in BSCS



V(G) = {2, 17, 56, 57, 11, 21, 22, 100, 123}
E(G) = {(17, 56), (56, 57), (57, 123), (17, 11), (11, 21), (21, 123), (11, 22), (22, 100), (2, 100)}

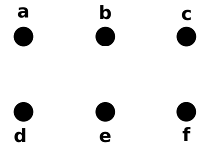
Graphs & Digraphs

- ***Example*:**
  - Draw the *directed* graph G where
    V(G) = {a, b, c, d, e, f}
    E(G) = {(a, d), (b, a), (b, e), (d, c), (f, e)}



Graph G

  - Draw the graph H where
    V(G) = {a, b, c, d, e, f}
    E(G) = {(a, d), (b, a), (b, e), (d, c), (f, e)}

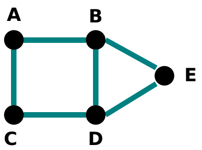


Graph H

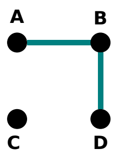
Subgraphs

- **SUBGRAPH of G**
  - **G_s = { $V(G_s)$, $E(G_s)$}**
    where $V(G_s) \subseteq V(G)$ and $E(G_s) \subseteq E(G)$

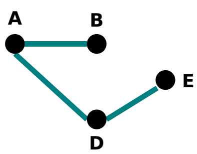
***Examples*:**



Graph G



Graph H1

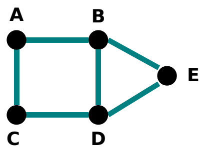


Graph H2

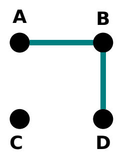
Subgraphs

- **SPANNING SUBGRAPH of G**
  - A subgraph **G_s = { $V(G_s)$, $E(G_s)$}** where $V(G_s) = V(G)$.

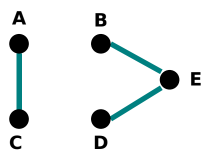
***Examples*:**



Graph G



Graph H1

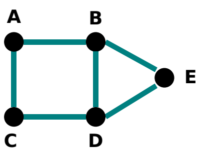


Graph H2

Subgraphs

- **SUBGRAPH INDUCED by a set of vertices W**
  - A subgraph **G_s = { $V(G_s)$, $E(G_s)$}** where $V(G_s)$=W and $E(G_s)$ are edges of G that join pairs of vertices in W.

***Examples*:**



Graph G

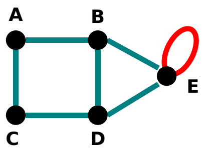
where W = {A,B,C,D}

where W = {A,B,D,E}

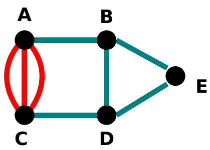
Graph Theory

- **SPECIAL EDGES**
  - **Loop** ≡ connects a vertex to itself
  - **Parallel/multiple edges** ≡ join same pair of vertices
  - **Weighted/ labeled** edges ≡ edges assigned weights

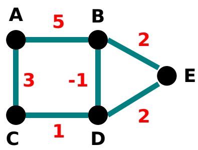
***Examples*:**



Graph G with loop



Graph H with parallel edges

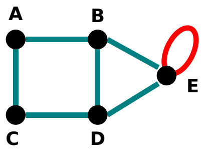


Graph J with weighted edges

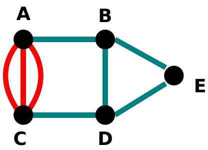
Graph Theory

- **SPECIAL EDGES**
  - **Multigraph** = graph with loops and/or parallel edges
  - **Simple graph** = graph w/o loops and/or parallel edges

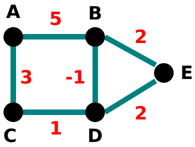
***Examples*:**



Graph G with loop



Graph H with parallel edges

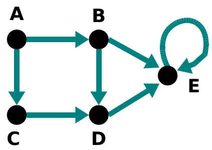


Graph J with weighted edges

Graph Theory

- **DEGREE of a VERTEX: Directed Graphs only**
  - **in-degree $\rho^+(v)$** $\equiv$ # of edges *incident to* v.
  - **out-degree $\rho^-(v)$** $\equiv$ # of edges *incident from* v.

***Examples*:**

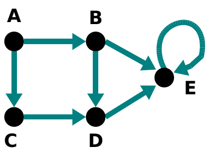


Graph G

Graph Theory

- **DEGREE of a VERTEX**
  - **Degree of vertex v $\rho(v)$** $\equiv$ # of edges *incident on* v.
    $\equiv$ **$\rho^-(v) + \rho^+(v)$** in directed graphs

***Examples*:**

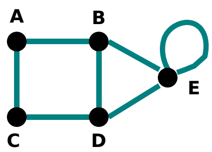


Graph G

Graph Theory

- **DEGREE of a VERTEX**
  - **Degree of vertex v $\rho(v)$** $\equiv$ # of edges *incident on* v.
    $\equiv$ **$\rho^-(v) + \rho^+(v)$** in directed graphs

***Examples*:**

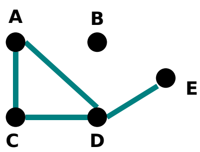


Graph G

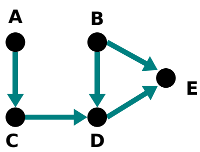
Graph Theory

- **SPECIAL VERTICES**
  - **Isolated vertex v** $\equiv \rho(v) = 0$.
  - **End vertex v** $\equiv \rho(v) = 1$.
  - **Sink vertex v** $\equiv \rho^+(v) = |V(G) - 1|$ and $\rho^-(v) = 0$.

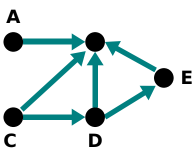
***Examples*:**



Graph G with isolated vertex



Graph H with end vertex

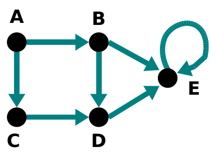


Graph J with sink vertex

Graph Theory

- **FIRST THEOREM of GRAPH THEORY**
  - sum of $\rho(v) = 2 \cdot |E(G)|$
  - sum of $\rho^+(v) = |E(G)|$
  - sum of $\rho^-(v) = |E(G)|$

***Example*:**

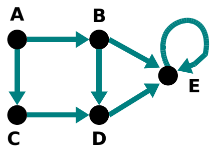


Graph G

Graph Theory

- **HANDSHAKING LEMMA**
  - # of vertices with odd degree ≡ even #

***Example*:**

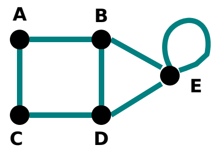


Graph G

Graph Representations

- **INCIDENCE MATRIX**
  - rows of incidence matrix M(G) ⇔ vertices
  - columns of incidence matrix M(G) ⇔ edges
  - entry for row v and column e = # of times e is incident on v.

***Example*:**



Graph G

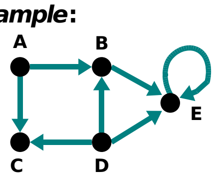
e_1 e_2 e_3 e_4 e_5 e_6 e_7

A
B
C
D
E

Graph Representations

- **INCIDENCE MATRIX**
  - rows of incidence matrix M(G) ⇔ vertices
  - columns of incidence matrix M(G) ⇔ edges
  - entry for row v and column e = # of times e is incident on v.

***Example*:**



Graph G

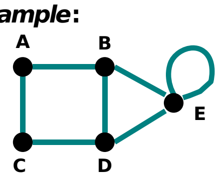
e_1 e_2 e_3 e_4 e_5 e_6 e_7

A
B
C
D
E

Graph Representations

- **ADJACENCY MATRIX**
  - rows of adjacency matrix M(G) ⇔ vertices
  - columns of adjacency matrix M(G) ⇔ vertices
  - entry for row i and column j = # of edges connecting i and j.

***Example*:**



Graph G

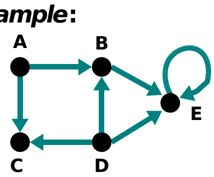
A B C D E

A
B
C
D
E

Graph Representations

- **ADJACENCY MATRIX**
  - rows of adjacency matrix M(G) ⇔ vertices
  - columns of adjacency matrix M(G) ⇔ vertices
  - entry for row i and column j = # of edges connecting i and j.

***Example*:**



Graph G

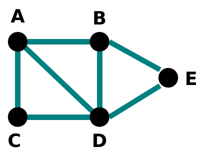
A B C D E

A
B
C
D
E

Graph Operations

- **Removal of a vertex v from graph G**
  - V(G – v) = V(G) – {v}
  - E(G –v) = E(G) except those incident on v

***Examples*:**



Graph G

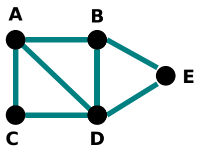
After removal of vertex A

After removal of vertex D

Graph Operations

- **Removal of a edge e from graph G**
  - V(G – e) = V(G)
  - E(G – e) = E(G) – {e}

***Examples*:**



Graph G

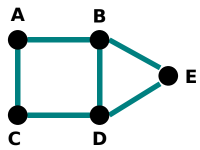
After removal of edge (A,D)

After removal of edge (D,E)

Graph Operations

- **Addition of an edge e to graph G**
  - V(G + e) = V(G)
  - E(G + e) = E(G) + {e}

***Examples*:**



Graph G

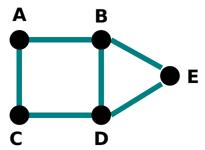
After adding edge (A,D)

After adding of edge (C,E)

Graph Operations

- **Complement of a (Simple) Graph**
  - $V(G^c) = V(G)$ and $E(G^c)$ = edges such that:
    - given vertices a,b $\in$ V(G):
    - edge (a,b) $\in E(G^c)$ iff edge (a,b) $\notin$ E(G)

***Examples*:**



Graph G

Graph G^c

Graph Theory

- **Outline**:
  - Introduction
    - Basic terminology and concepts
    - Representations of graphs
    - Operations on Graphs
  - **Walks, trails, paths, circuits and cycles**
    - Eulerian circuits
    - Hamiltonian cycles
  - Special types of graphs
  - Graph isomorphism and homeomorphism
  - Trees
  - Graph problems and their applications

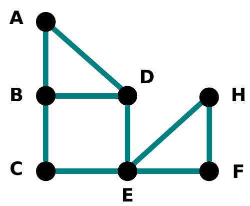
Walks, Trails, Paths, etc.

- **WALKS and related definitions**
  - **walk** $\equiv$ finite non-empty sequence of edges
    $(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n)$
    such that (v_i, v_{i+1}) is an edge in G.
      - denoted by $v_1\ v_2\ v_3 \ldots v_{n-1}\ v_n$.
      - v_1 is called the **initial vertex**
      - v_n is called the **final vertex.**
    - **length** $\equiv$ number of edges in the walk.
    - even number of edges $\equiv$ **even walk**
    - odd number of edges $\equiv$ **odd walk**
  - **trail** $\equiv$ walk + no repeated edges.
  - **path** $\equiv$ walk + no repeated vertices.

Walks, Trails, Paths, etc.

- **WALKS and related definitions**

***Example*:**



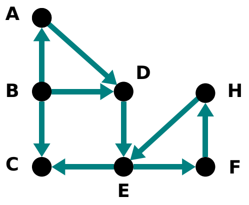
Graph G

ABCBDEH
ABEDHFE
FHEDBC
ABDECBA
ABDA

Walks, Trails, Paths, etc.

- **WALKS and related definitions**

***Example*:**



Graph G

ADEFHEC
BADEFHEF
ADBCE
ADEC

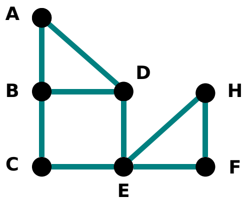
Walks, Trails, Paths, etc.

- **WALKS and related definitions**
  - **closed walk**
    ≡ walk that begins and ends at the same vertex.
  - **closed trail** or **circuit**
    ≡ closed walk + no repeated edges.
  - **closed path** or **cycle**
    ≡ closed walk + no repeated vertices.

Walks, Trails, Paths, etc.

- **WALKS and related definitions**

***Example*:**



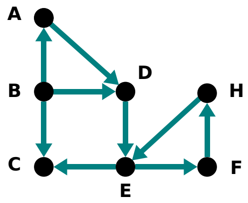
Graph G

ADECBDA
BCEDAB
BDEHFECB
ABDEFHEDA

Walks, Trails, Paths, etc.

- **WALKS and related definitions**

***Example*:**



Graph G

EFHE
BDECB
FHEF

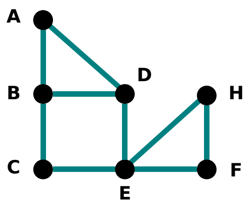
Walks, Trails, Paths, etc.

- **CONNECTED and DISCONNECTED GRAPHS**
  - **connected (undirected) graph**
    ≡ there is a path between <u>any two</u> of its vertices.
  - **disconnected (undirected) graph**
    ≡ a (undirected) graph that is not connected.
  - **components** ≡ connected subgraphs of a graph
    (*Note*: A connected graph is therefore made up of only one component.)
  - **number of components** denoted by **C(G)**

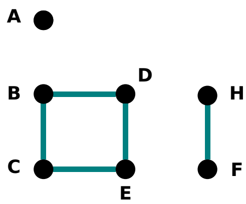
Walks, Trails, Paths, etc.

- **CONNECTED and DISCONNECTED GRAPHS**

***Example*:**



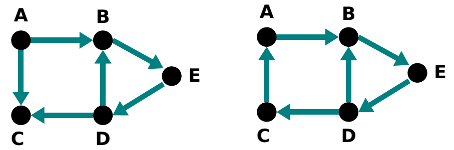
Graph G



Graph H

Walks, Trails, Paths, etc.

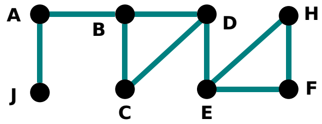
- **STRONGLY CONNECTED GRAPHS**
  - **strongly connected (directed) graph**
    ≡ there is a path between <u>any two</u> of its vertices.
  - **weakly connected (directed) graph**
    ≡ there is a path between any two of its vertices in the *underlying undirected graph*.



Graph G Graph H

Walks, Trails, Paths, etc.

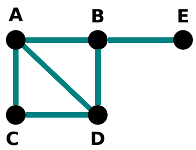
- **CUT EDGES and CUT VERTICES**
  - **cut edge** of graph G
    ≡ edge e such that G – e is a disconnected graph.
  - **cut set** ≡ set of all cut edges in G.
  - **cut vertex/point** of a graph G
    ≡ vertex v such that G – v is a disconnected graph.
  - THEOREM. An edge e of a graph G is a cut edge of G iff it is not contained in any cycle of the graph G.



Graph G

Walks, Trails, Paths, etc.

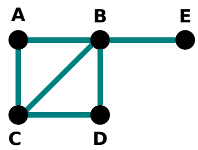
- **DISTANCES and DIAMETERS**
  - **distance between two vertices** u and v, **d(u,v)**
    ≡ length of the *shortest path* between u and v.
  - **diameter of a connected graph**
    ≡ *maximum distance* between any two of its vertices.



Graph G

Walks, Trails, Paths, etc.

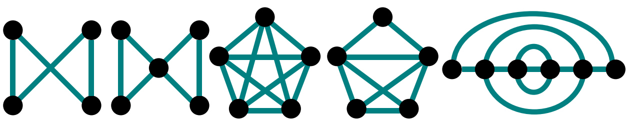
- **DISTANCES and DIAMETERS**
  - **distance between two vertices** u and v, **d(u,v)**
    ≡ length of the *shortest path* between u and v.
  - **diameter of a connected graph**
    ≡ *maximum distance* between any two of its vertices.



Graph H

EULERIAN CIRCUITS

- **Eulerian circuit** in a graph G
  ≡ circuit which includes all vertices and all edges of G.
- **Eulerian** graph ≡ if it contains a Eulerian circuit.
- THEOREM
  A non-empty connected graph is Eulerian if and only if and only if every vertex has an even degree.

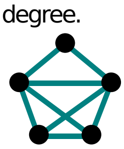


Graph G Graph H Graph J Graph K Graph L

EULERIAN CIRCUITS

Remarks:

- If we can find a trail through a graph that covers every edge exactly once but does not begin and end at the same vertex, then the graph is simply **edge-traceable**.
- A graph is edge-traceable if and only if it has *at most* two vertices of odd degree.

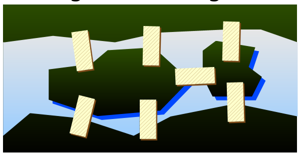


Graph K

EULERIAN CIRCUITS

An application: ***Königsberg Bridge Problem***

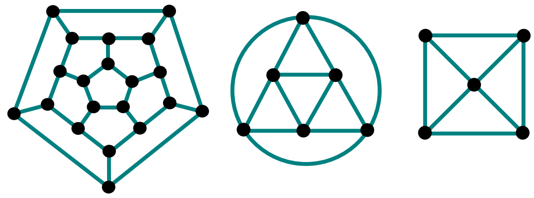
- Pass through all the bridges and return to starting point.



- Need to find an Eulerian circuit in corresponding graph
  - Vertices = locations
  - Edges = bridges

HAMILTONIAN CYCLES

- **Hamiltonian cycle**
  ≡ a cycle which includes every vertex in the graph G exactly once (except for the initial and final vertices).
- A connected graph is **Hamiltonian** if it contains a Hamiltonian cycle.



Graph Theory

- **Outline**:
  - **Introduction**
    - Basic terminology and concepts
    - Representations of graphs
    - Operations on Graphs
  - Walks, trails, paths, circuits and cycles
    - Eulerian circuits
    - Hamiltonian cycles
  - Special types of graphs
  - Graph isomorphism and homeomorphism
  - Trees
  - Graph problems and their applications