Graphs-Introduction

November 9, 2016

Graphs - Basic Concepts

- Basic definitions: vertices and edges
- More definitions: paths, simple paths, cycles, loops
- Connected and disconnected graphs
- Spanning trees
- Complete graphs
- Weighted graphs and networks
- Graph representations
  - Adjacency matrix
  - Adjacency lists

Graph

A graph is a mathematical object that can be used to model many problems – objects and processes:

- Linked list
- Tree (particular instance of a graph)
- Flowchart of a program
- Road map
- Electric circuits
- Course curriculum
- Communication Network
- Friend-Follower Graph (Twitter)
- Facebook
- Internet

Vertices and Edges

Definition: A graph ($\mathcal{G}$) is a collection (nonempty set) of vertices and edges

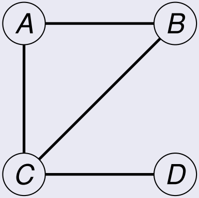
- Vertices ($\mathcal{V}$):
  - can have names and properties
- Edges ($\mathcal{E}$):
  - connect two vertices,
  - can be labeled,
  - can be directed
  - Edge set

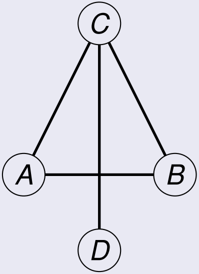
$$\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$$

  - Adjacent vertices: there is an edge between them
- We write: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

Example

- Vertex set : $\mathcal{V} = \{A, B, C, D\}$
- Edge set : $\mathcal{E} = \{(A, B), (A, C), (B, C), (C, D)\}$

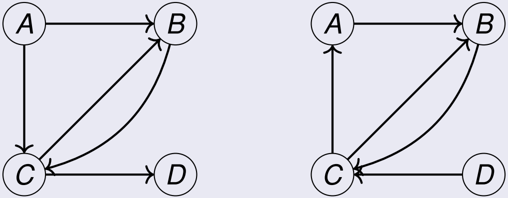




Two ways to draw the same graph

Directed versus Undirected

Directed Graph: each edge is associated with an ordered pair of vertices.

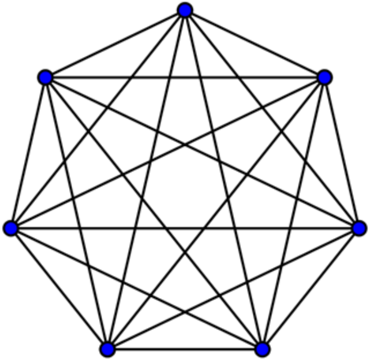


Different Graphs

A complete undirected unweighted graph

is one where there is an edge connecting all possible pairs of vertices in a graph.
The complete graph with n vertices is denoted as K_n.

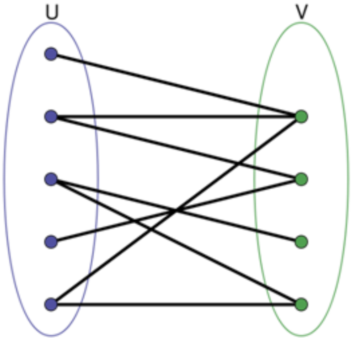
- Dense graphs: relatively few of the possible edges are missing
- Sparse graphs: relatively few of the possible edges are present



Bipartite Graph

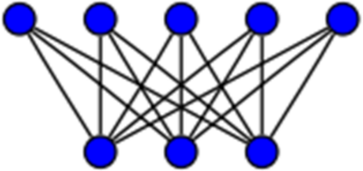
A graph is bipartite

- if there exists a way to partition the set of vertices V, in the graph into two sets V_1 and V_2
- where $V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \emptyset$, such that each edge in E contains one vertex from V_1 and the other vertex from V_2.



Complete bipartite graph

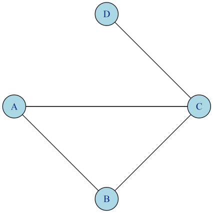
A complete bipartite graph on m and n vertices is denoted by $K_{m,n}$ and consists of $m + n$ vertices, with each of the first m vertices connected to all of the other n vertices, and no other vertices.



More definitions : Path

- A path
  - A path of length n from vertex v_0 to vertex v_n is an alternating sequence of $n + 1$ vertices and n edges beginning with vertex v_0 and ending with vertex v_n in which edge e_i is incident upon vertices v_{i-1} and v_i.
    - (The order in which these are connected matters for a path in a directed graph in the natural way.)
- A connected graph
  - A connected graph is one where any pair of vertices in the graph is connected by at least one path.

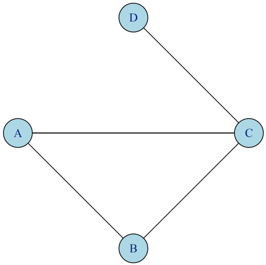
A B C
B A C D
A B C A B C A B C D
B A B A C



More definitions : Cycle

Simple path with distinct edges, except that the first vertex is equal to the last

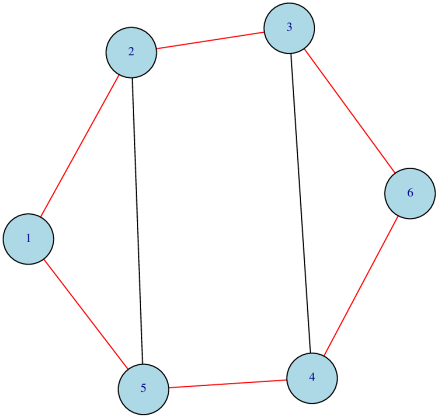
A B C A
B A C B
C B A C



A graph without cycles is called acyclic graph.

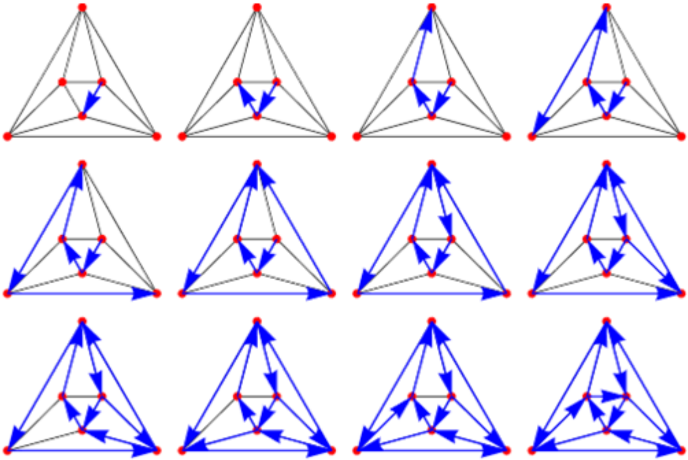
More definitions : Hamiltonian Cycle

- A Hamiltonian cycle
  - A Hamiltonian cycle is a simple cycle that contains all the vertices in the graph



More definitions : Euler Cycle

- A Euler cycle
  - An Euler cycle is a cycle that contains every edge in the graph exactly once.
    - Note that a vertex may be contained in an Euler cycle more than once. Typically, these are known as Euler circuits, because a circuit has no repeated edges

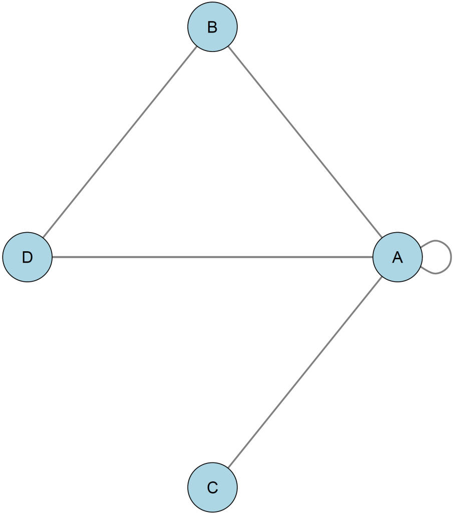


Euler Circuit vs Hamiltonian Cycle

- There is a nice simple method for determining if a graph has an Euler circuit
- but no such method exists to determine if a graph has a Hamiltonian cycle.
- The latter problem is an NP-Complete problem.
  - In a nutshell, this means it is most-likely difficult to solve perfectly in polynomial time.

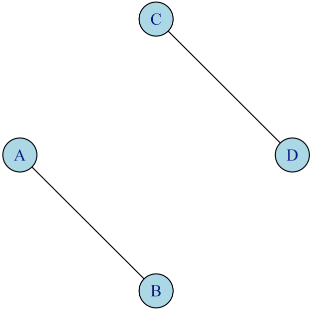
More definitions : Loop

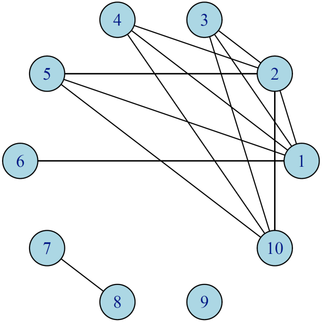
- Loop: An edge that connects the vertex with itself



Connected and Disconnected graphs

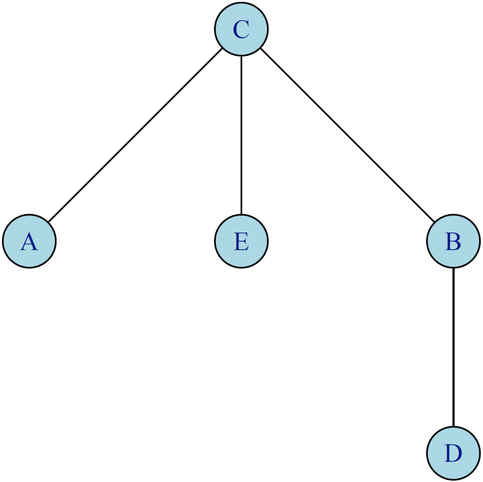
- Connected graph: There is a path between each two vertices
- Disconnected graph : There are at least two vertices not connected by a path.
- Examples of disconnected graphs:



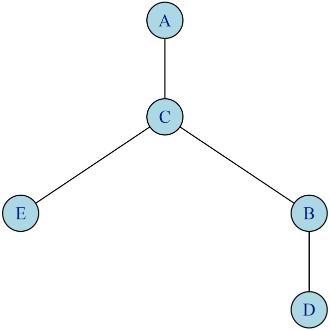


Graphs and Trees

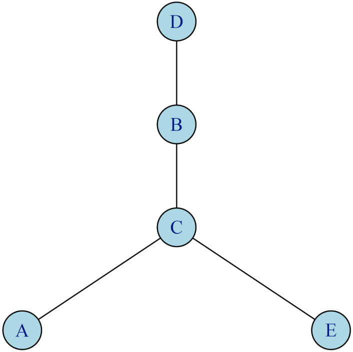
- Tree: an undirected graph with no cycles, and a node chosen to be the root



Graphs and Trees



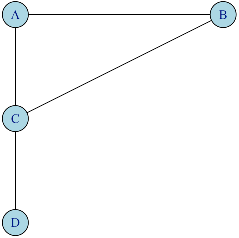
Root A



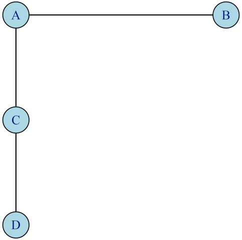
Root D

A spanning tree of an undirected graph

- Spanning tree: A sub-graph that contains all the vertices, and no cycles.
- If we add any edge to the spanning tree, it forms a cycle, and the tree becomes a graph



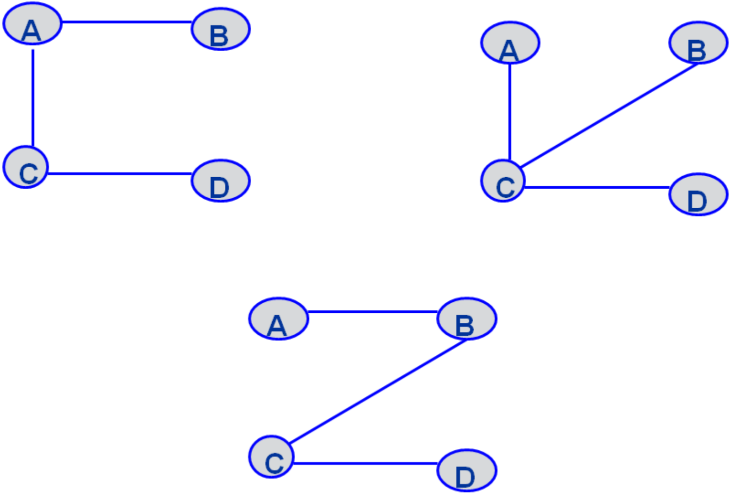
Graph



Spanning Tree

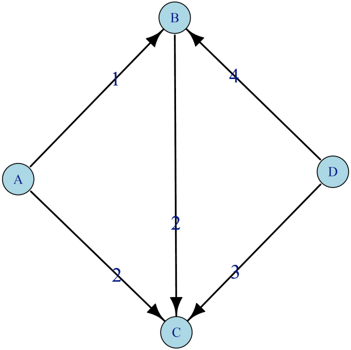
Examples

All spanning trees of the graph on the previous slide



Weighted graphs and Networks

- Weighted graphs – weights are assigned to each edge (e.g. road map)
- Networks: directed weighted graphs (some theories allow networks to be undirected)

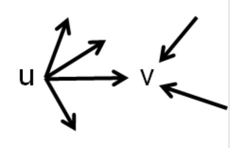


Some Graph Terminology

- u is the source , v is the sink of (u,v)

  u $\rightarrow$ **v**

  b __ **c**
- u, v, b, c are the endpoints of (u,v) and (b, c)
- u, v are adjacent nodes. b, c are adjacent nodes



- outdegree of u in directed graph: number of edges for which u is source
- indegree of v in directed graph: number of edges for which v is sink
- degree of vertex w in undirected graph: number of edges of which w is an endpoint

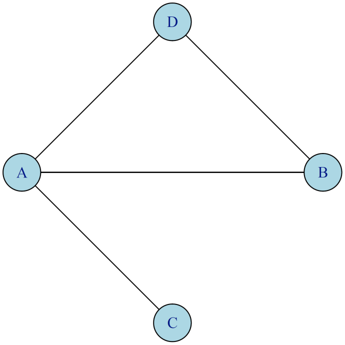
Graph Representation

- Adjacency matrix
- Adjacency lists

Adjacency matrix – undirected graphs

Vertices: A,B,C,D
Edges: AC, AB, AD, BD

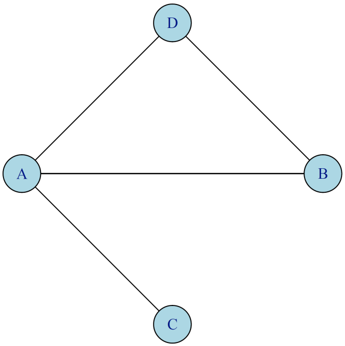
	A	B	C	D
A	0	1	1	1
B	1	0	0	1
C	1	0	0	1
D	1	1	0	0



Adjacency matrix – undirected graphs

Vertices: A,B,C,D
Edges: AC, AB, AD, BD
The matrix is symmetrical

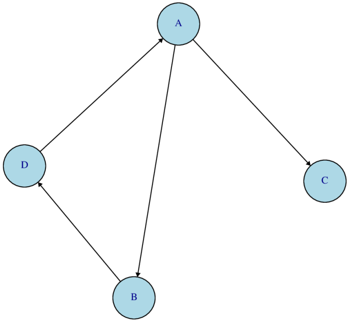
	A	B	C	D
A	0	1	1	1
B	1	0	0	1
C	1	0	0	1
D	1	1	0	0



Adjacency matrix – directed graphs

Vertices: A,B,C,D
Edges: AC, AB, BD, DA

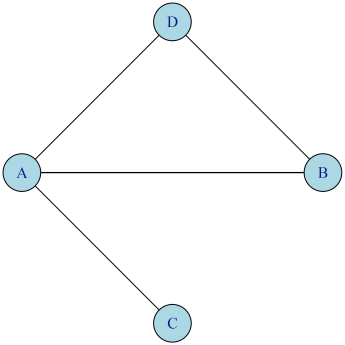
	A	B	C	D
A	0	1	1	0
B	0	0	0	1
C	0	0	0	0
D	1	0	0	0



Adjacency lists – undirected graphs

Vertices: A,B,C,D
Edges: AC, AB, AD, BD

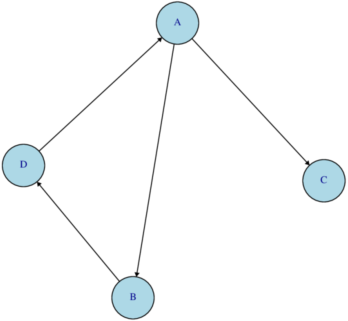
Heads	Lists
A	B C D
B	A D
C	A
D	A B



Adjacency lists – directed graphs

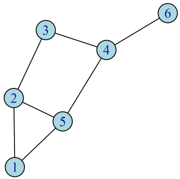
Vertices: A,B,C,D
Edges: AC, AB, BD, DA

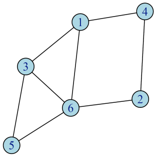
Heads	lists
A	B C
B	D
C	-
D	A



The complement of a graph

- The complement of a graph G is a graph G' which contains all the vertices of G, but for each edge that exists in G, it is NOT in G', and for each possible edge NOT in G, it IS in G'.





	1	**2**	**3**	**4**	**5**	**6**
1	*0*	***1***	*0*	*0*	***1***	*0*
2	***1***	*0*	***1***	*0*	***1***	*0*
3	*0*	*1*	*0*	***1***	*0*	*0*
4	*0*	*0*	*1*	*0*	***1***	***1***
5	*1*	*1*	*0*	*1*	*0*	*0*
6	*0*	*0*	*0*	*1*	***0***	***0***

	1	**2**	**3**	**4**	**5**	**6**
1	*1*	*0*	*1*	*1*	*0*	*1*
2	*0*	*1*	*0*	*1*	*0*	*1*
3	*1*	*0*	*1*	*0*	*1*	*1*
4	*1*	*1*	*0*	*1*	*0*	*0*
5	*0*	*0*	*1*	*0*	*1*	*1*
6	*1*	*1*	*1*	*0*	*1*	*1*

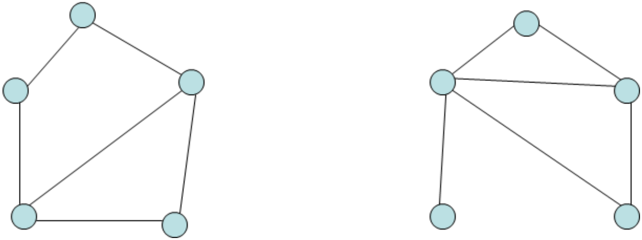
Graph Isomorphisms

- Definition: Let $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ be simple graphs. The graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are isomorphic iff
  - There exists a bijection $f : \mathcal{V}_1 \to \mathcal{V}_2$
  - if for all v_1 and v_2 in $\mathcal{V}_1$
    if v_1 and v_2 are adjacent in $\mathcal{G}_1$
    then
    $f(v_1)$ and $f(v_2)$ are adjacent in $\mathcal{G}_2$
- Determining if two graphs are isomorphic or not is a hard problem. One needs to come up with the isomorphism (the bijection f) in order to show that they are isomorphic.

Invariants

- Invariants: things that $\mathcal{G}_1$ and $\mathcal{G}_2$ must have in common to be isomorphic:
  - The same number of vertices
  - The same number of edges
  - Degrees of the corresponding vertices are the same
  - If one is bipartite, the other must also be
  - If one is complete, the other must also be
- If two graphs have different invariants, then, they cannot be isomorphic !

Example



- Are they isomorphic?
- Same number of vertices, same number of edges, but . . .