Graphs and Digraphs

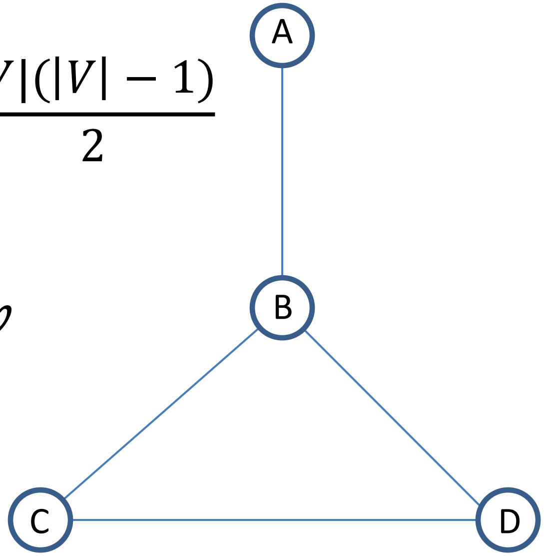
A World of Networks

Summary

- Basic concepts related to graphs and digraphs
- Examples of problems in graphs and digraphs
- Euler circuits and an algorithm to find them

Graphs: basic definitions

$$|E| \leq \frac{|V|(|V|-1)}{2}$$



- Graph $G = (V, E)$
  - Set V of *vertices* or *nodes*
  - Set E of *edges*; two-element subsets of V; $\{u, v\}$ rep. by uv
  - If $\{u, v\} \in E$, then u and v are its *ends*; the edge *joins* u and v; u and v are *neighbors;* edge $\{u, v\}$ is incident on u and v
  - Set $E(v)$ of neighbors of v; $|E(v)|$ is the degree of v

Handshaking theorem: $\sum_{v \in V} |E(v)| = 2|E|$

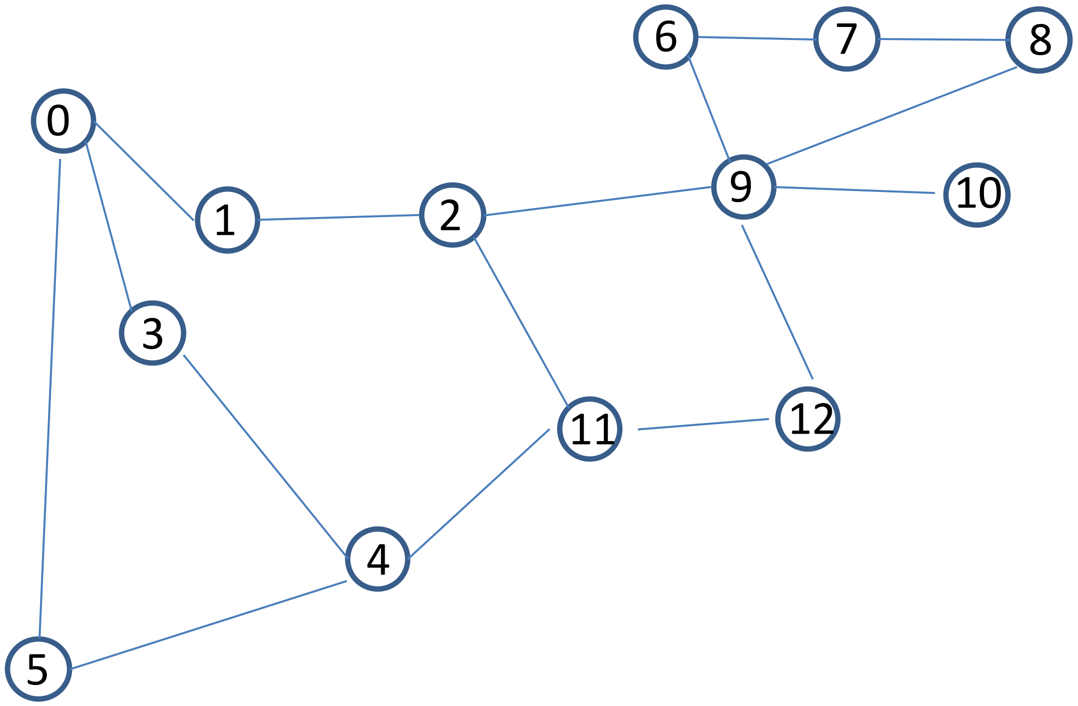
Every graph has an even number of nodes of odd degree

Every graph has two nodes with the same degree

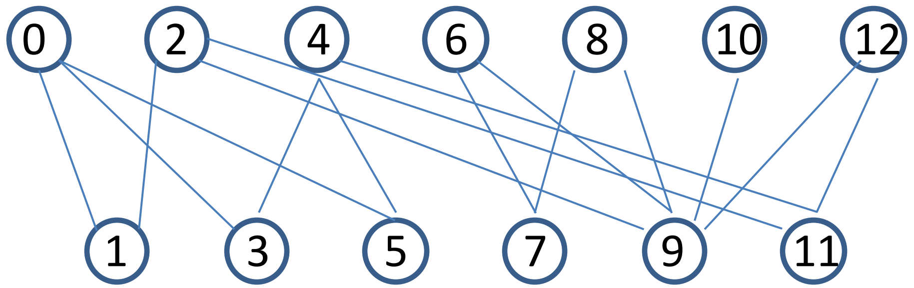
Graphic representation of a graph

Definition: A graph is bipartite if the set of nodes can be partitioned in two classes such that every edge has its ends in different classes

Is the following graph bipartite?



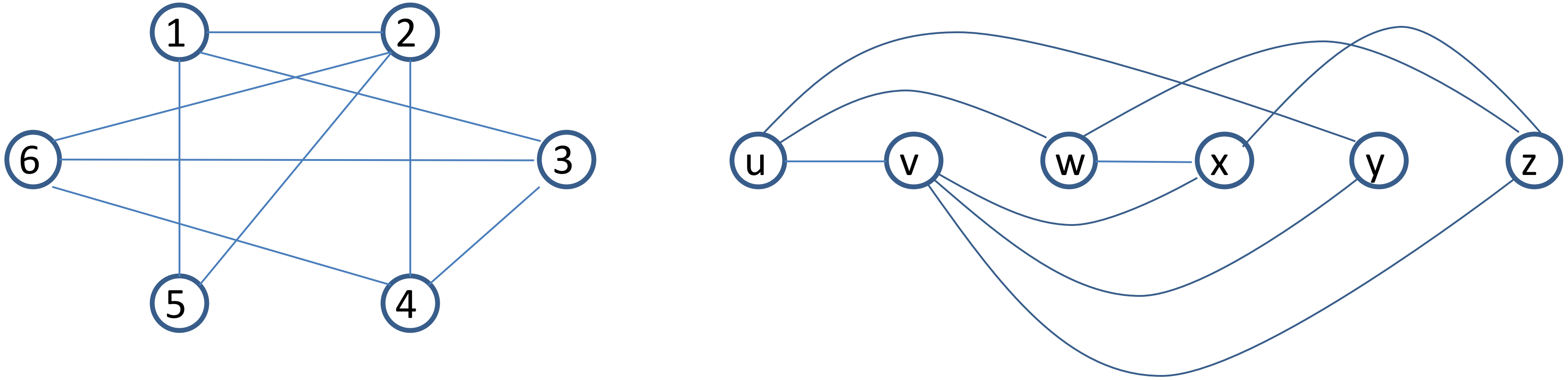
Yes, clearly!



How can a bipartite graph be characterized in terms of its cycles?

Graph isomorphism (node re-labeling)

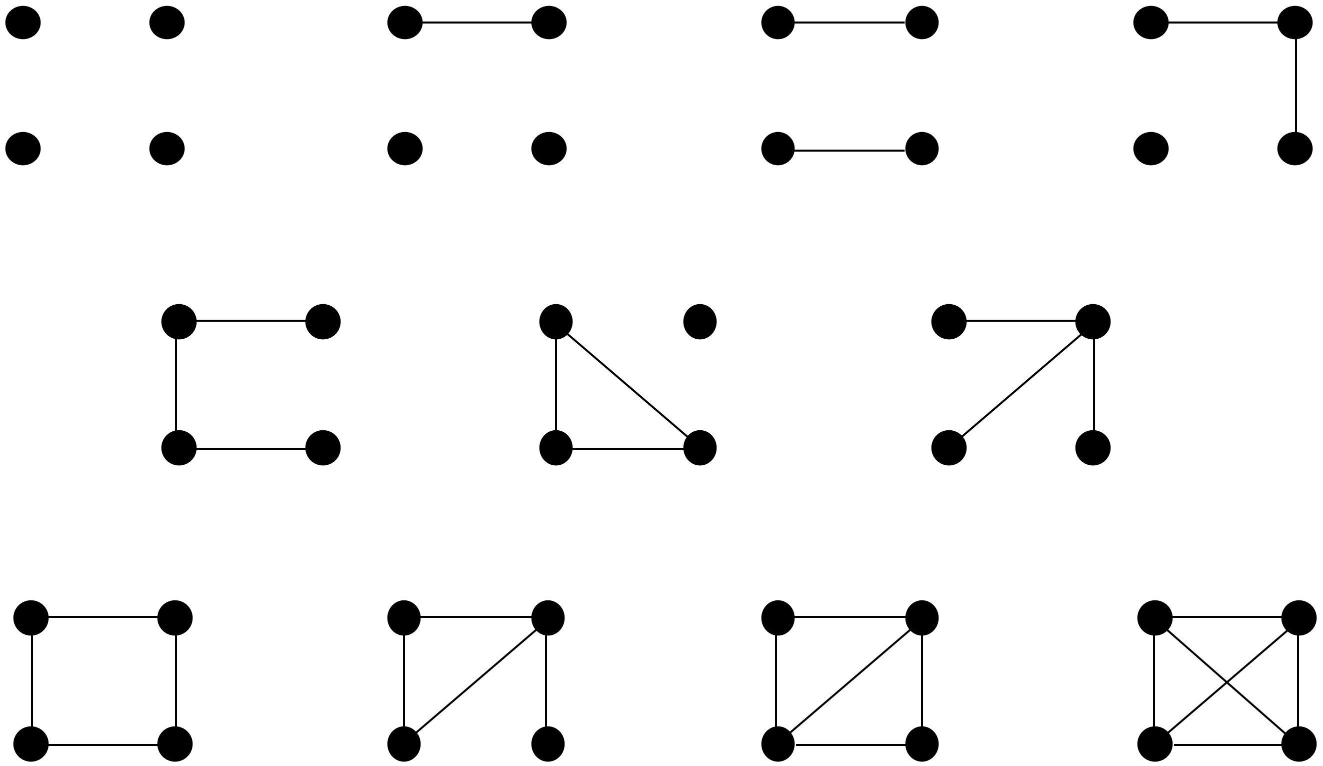
$G = (V, E)$ and $G' = (V', E')$ are isomorphic if there a bijection $f : V \to V'$ such that $\{u, v\} \in E$ if and only if $\{f(u), f(v)\} \in E'$



$$f(1) = u, f(2) = v, f(3) = w, f(4) = x, f(5) = y, f(6) = z$$

Graph isomorphism is an equivalence relation

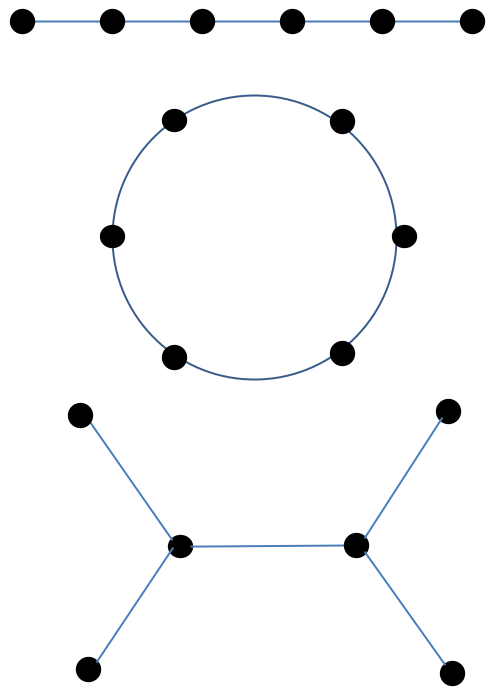
Isomorphism classes



$2^{(4\times3)/2} = 64$ *graphs, but only* 11 *isomorphism classes*

Paths, cycles, and trees

- Simple path
  - $V = \{v_0, v_1, \ldots, v_n\}$
  - $E = \{v_0v_1, v_1v_2, \ldots, v_{n-1}v_n\}$
- Cycle
  - $V = \{v_0, v_1, \ldots, v_{n-1}\}$
  - $E = \{v_0v_1, v_1v_2, \ldots, v_{n-2}v_{n-1}, v_{n-1}v_0\}$
- Tree
  - Connected and acyclic graph



***Proposition**: The following are equivalent characterizations of trees:*
1. *Connected graph with minimum number of edges;*
2. *Acyclic graph with maximum number of edges;*
3. *Connected or acyclic, and number of edges is one less the number of nodes;*
4. *Unique path between any two nodes.*

Walk, trail, and circuit in a graph

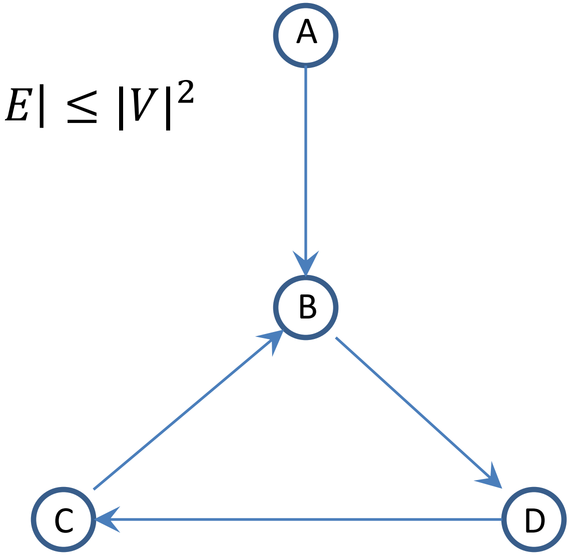
- Path, or walk, in G
  - Sequence $v_0 v_1 v_2 \cdots v_n$ of nodes of G such that consecutive nodes joined by an edge of G
- Trail in G
  - Walk without repeat edges
- Circuit in G
  - A closed trail without repeated edges

Every cycle with n nodes harbors n distinct circuits in each direction, each starting and ending at a different node of the cycle.

Digraphs: basic definitions

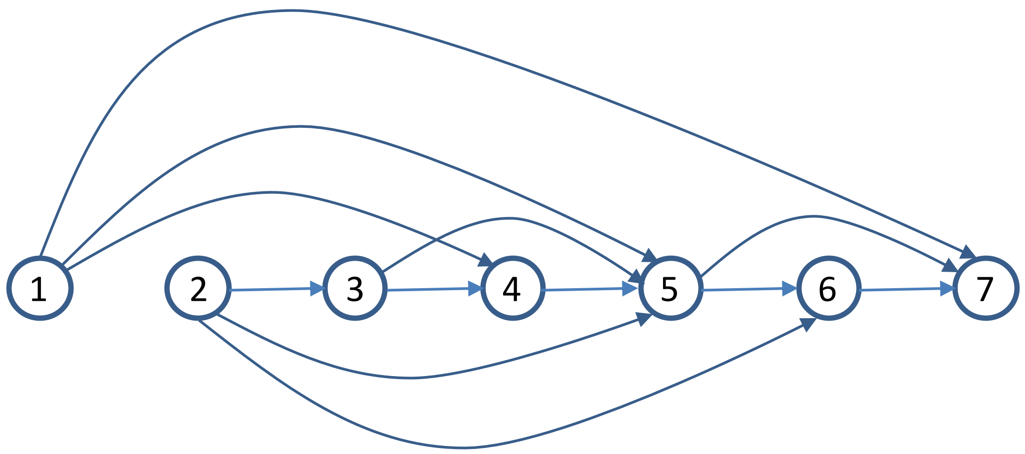
- Digraph $G = (V, A)$
  - Set V of *vertices* or *nodes*
  - Set A of *arcs* or *links*; subset of $V \times V$; (u, v) rep. by uv
  - If $(u, v) \in A$, then u is its *tail* and v is its *head*; u is an *in-neighbor* of v; v is an *out-neighbor* of u; the link *leaves* u and *enters* v; the link is *outgoing* from u and *incoming* to v
  - Sets $E^-(v)$ and $E^+(v)$ of in-neighbors and out-neighbors of v, respectively

$|E| \leq |V|^2$



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

Graphs and digraphs are structurally different



Digraph has 7 nodes, 12 links, and is acyclic

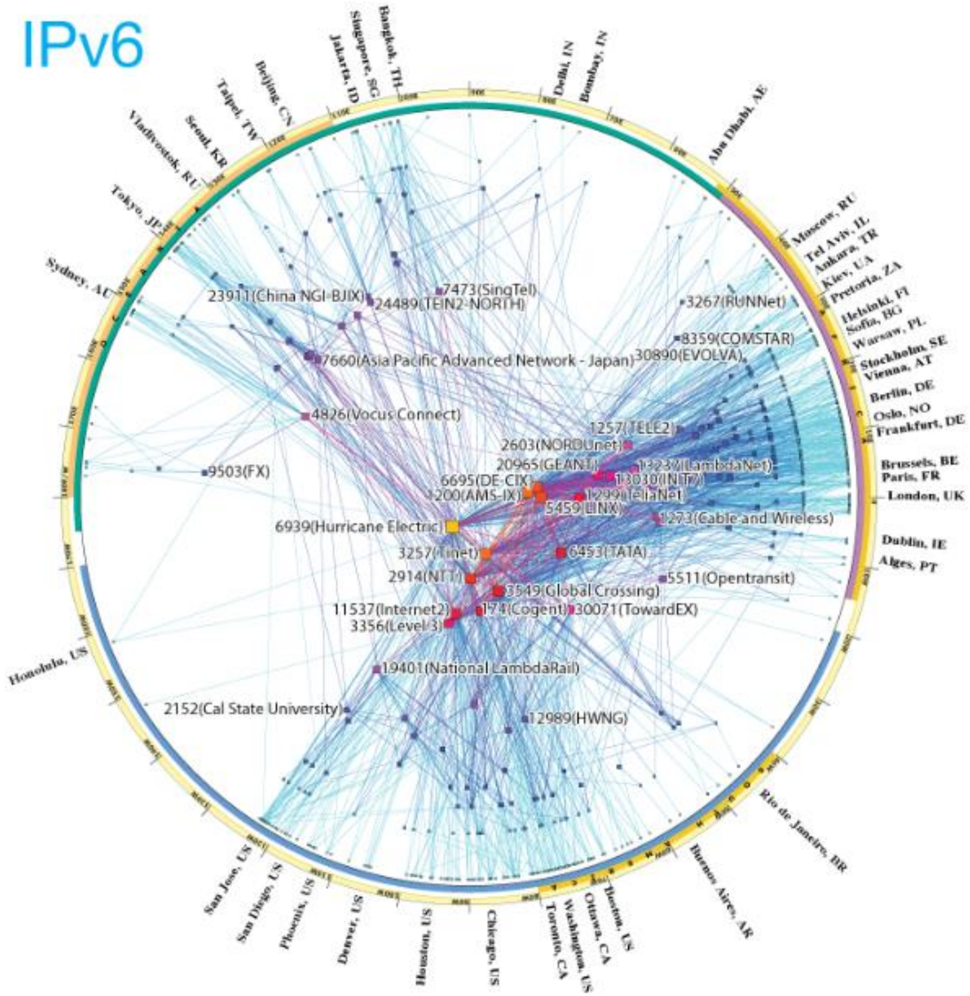
On the other hand, any graph with 7 nodes and 12 edges must contain at least one cycle

1. *What is the maximum number of edges in an acyclic graph?*
2. *What is the maximum number of arcs in an acyclic digraph (DAG)?*

National electrical grid

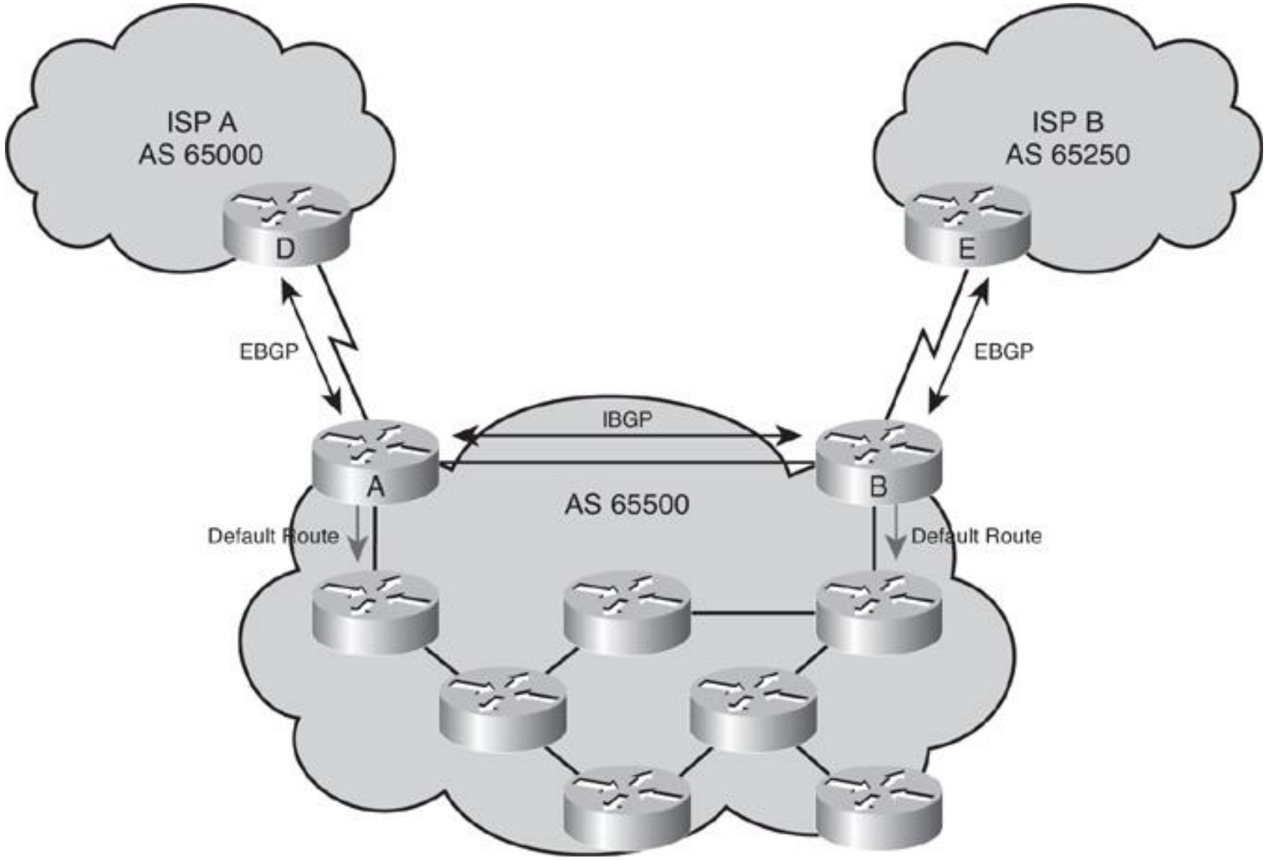
What is the impact of a failure of a sub-station?

IPv6 AS-level graph



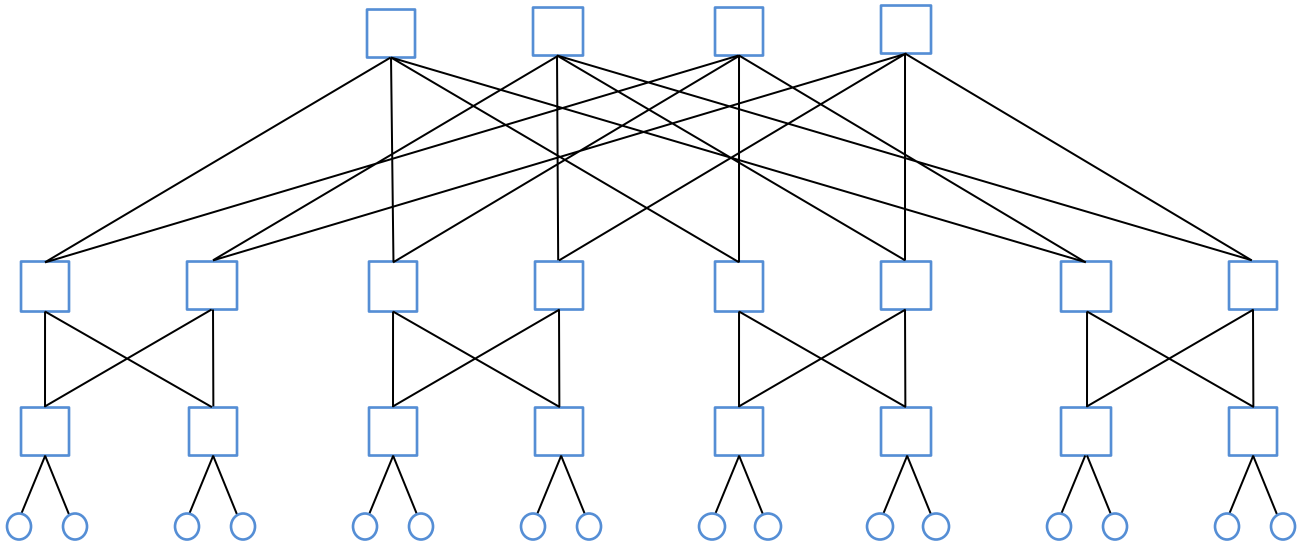
Is the IPv6 AS-level graph connected? Is it connected in terms of valid paths?

Router-level graph



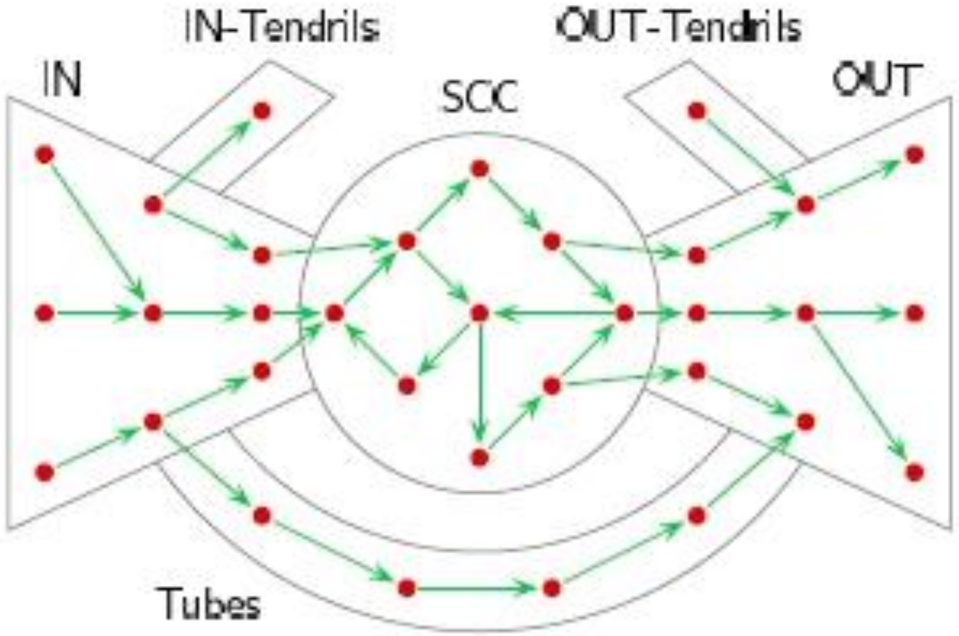
Are the router-level paths optimal? What is the optimality criterion?

Data Centers



Can we connect all sources to different destinations at the same time by disjoint paths?

World Wide Web

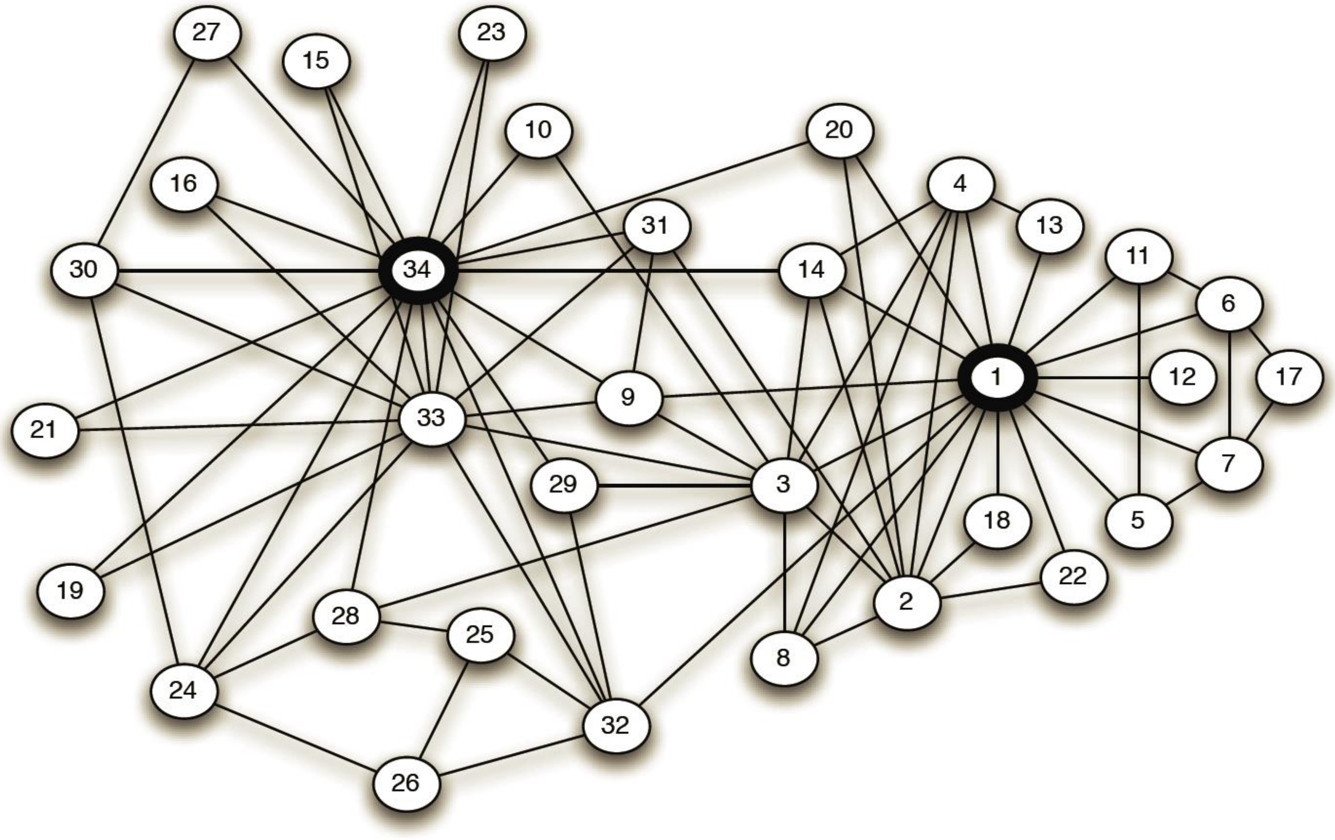


What is the size of the largest strongly connected component?

Lisbon underground

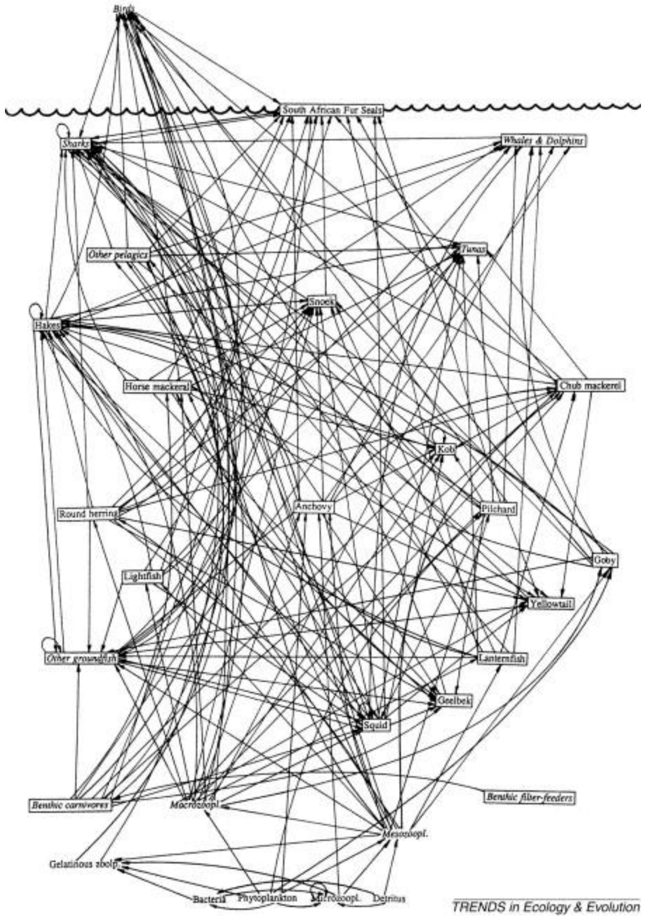
How to travel with minimum number of stops? Minimum number of transfers?

Social networks



Can you predict how the group will split?

Fishery food web



What should you hunt to increase the population of cod?

Dating in a particular high school

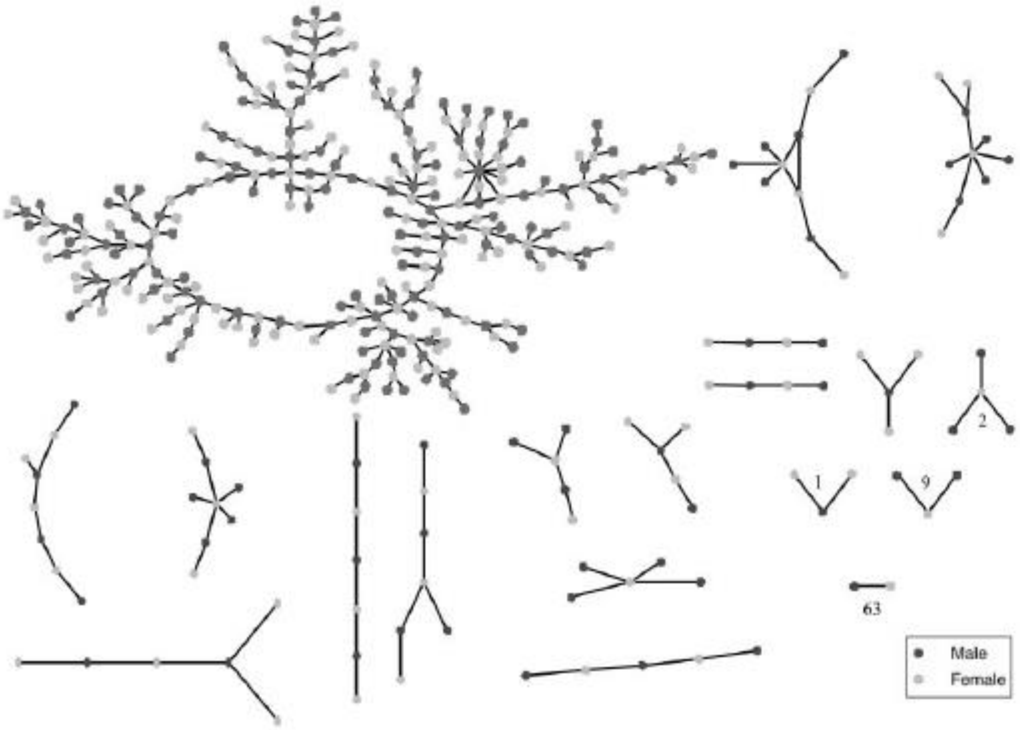
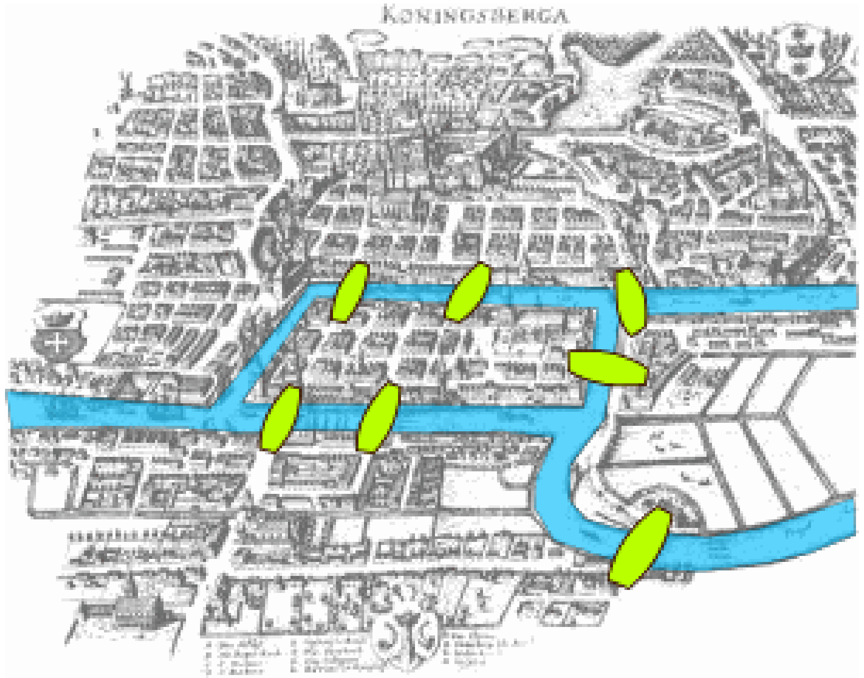
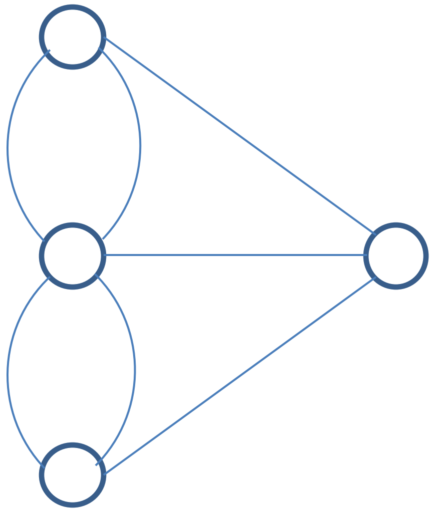


FIG. 2.—The direct relationship structure at Jefferson High

Did my girlfriend's ex-boyfriend dated my ex-girlfriend?

Seven bridges of Konigsberg





Is it possible to find a circuit through town that crosses each bridge exactly once?

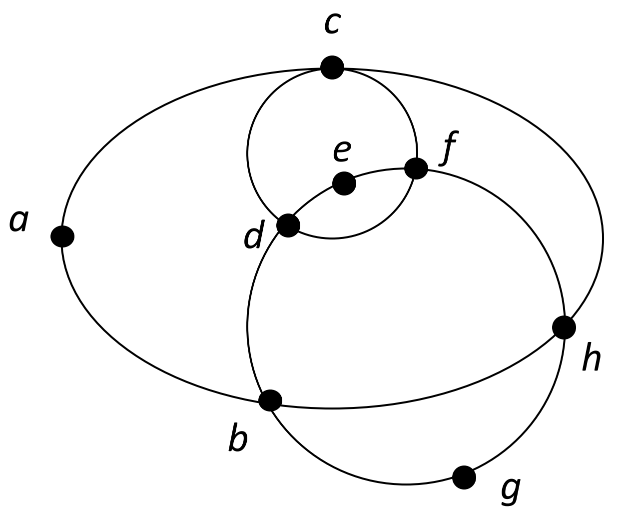
Euler circuit

Definition: An Euler circuit is a closed path the traverses all edges of a graph exactly once.

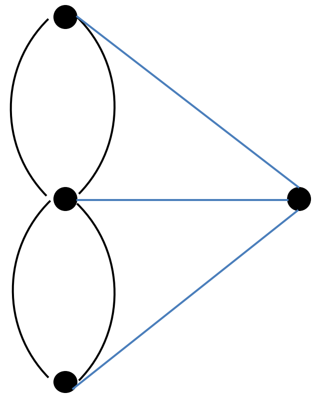
Proposition: A connected multigraph has an Euler circuit if, and only if, every node has even degree.

The Euler circuit problem is related to the Chinese postman problem, which asks for the shortest closed walk that visits all edges of a graph.

Examples



Euler circuit: *abhcdfhgbdefca*



No Euler circuit

Euler circuit – necessity of even degree

Proposition: A connected multigraph has an Euler circuit if, and **only if**, every node has even degree.

Proof (sketch): Other than the origin, a node visited k times ($k \geq 1$) has degree $2k$; the origin is visited k times ($k \geq 0$) has degree $2k + 2$.

Verification that at least one node has odd degree leads to the conclusion that there is no Euler circuit

Euler circuit – sufficiency of even degree

Proposition: A connected multigraph has an Euler circuit **if**, and only if, every node has even degree

Proof (sketch, contradiction): Let $T = v_0 e_0 \cdots e_{k-1} v_k$ be a longest path that does not repeat an edge in the multi-graph. We show that T is an Euler circuit.

1. Since T cannot be extended, it contains all edges incident at v_k. Because the degree of v_k is even, it must be the case that $v_k = v_0$, so that T is a circuit.
2. If T is not an Euler circuit, then there is an edge $e = uv_i$ outside T, but incident in a node v_i of T. Then, path $uev_i e_i \cdots e_{k-1} v_k e_0 \cdots e_{i-1}\, v_i$ does not repeat edges and is longer than T, which contradicts the definition of T.

The proof gives no clue on how to design an efficient algorithm to compute an Euler circuit!

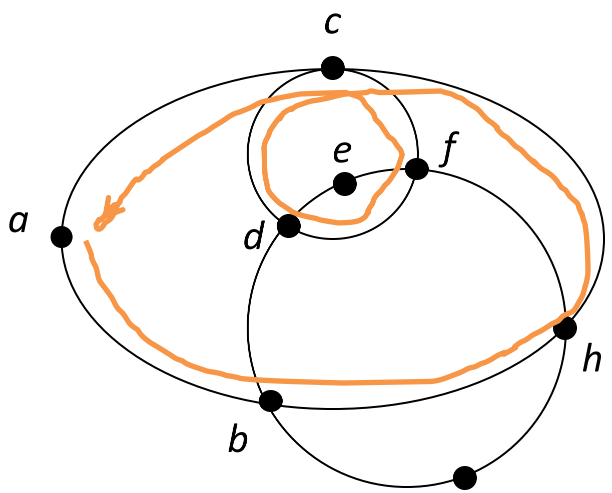
Polynomial-time algorithm for Euler circuit

Input: connected multi-graph $G = (V, E)$ where every node has even degree
Output: Euler circuit
Variables: stacks *head* and *tail;* at termination, *head* will be empty and *tail* will contain the Euler circuit

Pseudo-code
head := x; *tail* = ∅ /* *x* is an arbitrary node*/
1. While *head* is not empty
   a. While top node *v* of *head* is not isolated /* find a circuit */
      - Choose an edge *vw* and remove it from the graph
      - Push *w* to *head*
   b. While top node *v* of *head* is isolated and *head* is not empty /* backtrack to find another circuit */
      - Pop *v* from *head*
      - Push *v* to *tail*
2. Euler circuit can be found in *tail*

The algorithm runs in O(|E|) time, asymptotically as efficient as it can be!

Algorithm for Euler circuit: example

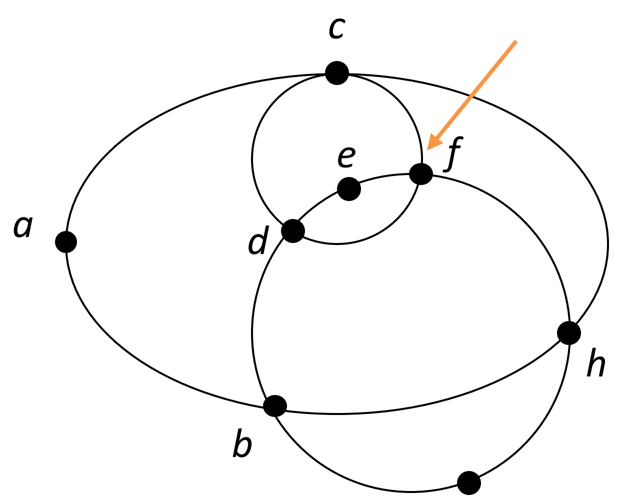


Iter.	head	tail
0	*a*	∅
1—7	*a* *c* *f* *d* *c* *h* *b* *a*	∅

While top node *v* of *head* is not isolated

- Choose an edge *vw* and remove it from the graph
- Push *w* to *head*

Algorithm for Euler circuit: example

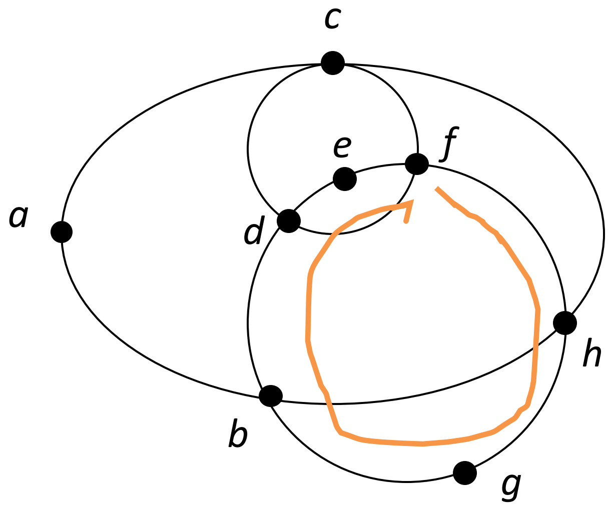


Iter.	head	tail
0	*a*	∅
1—7	*a* *c* *f* *d* *c* *h* *b* *a*	∅
8—9	*f* *d* *c* *h* *b* *a*	*c* *a*

While top node *v* of *head* is isolated and *head* is not empty

- Pop *v* from *head*
- Push *v* to *tail*

Algorithm for Euler circuit: example



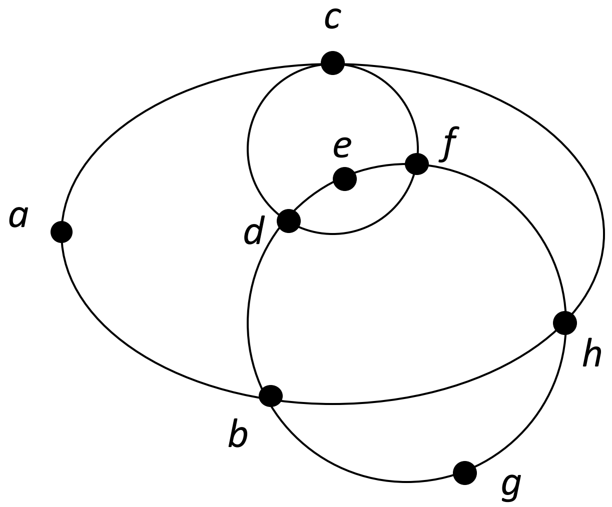
Iter.	head	tail
0	*a*	∅
1—7	*a* *c* *f* *d* *c* *h* *b* *a*	∅
8—9	*f* *d* *c* *h* *b* *a*	*c* *a*

Iter.	head	tail
10—15	*f* *e* *d* *b* *g* *h* *f* *d* *c* *h* *b* *a*	*c* *a*

While top node *v* of *head* is not isolated

- Choose an edge *vw* and remove it from the graph
- Push *w* to *head*

Algorithm for Euler circuit: example



While top node *v* of *head* is isolated and *head* is not empty

- Pop *v* from *head*
- Push *v* to *tail*

Iter.	head	tail
0	*a*	Ø
1—7	*a* *c* *f* *d* *c* *h* *b* *a*	Ø
8—9	*f* *d* *c* *h* *b* *a*	*c* *a*

Iter.	head	tail
10—15	*f* *e* *d* *b* *g* *h* *f* *d* *c* *h* *b* *a*	*c* *a*

Iter.	head	tail
16—27	Ø	*a* *b* *h* *c* *d* *f* *h* *g* *b* *d* *e* *f* *c* *a*

Euler circuit: abhcdfhgbdefca

Hamilton cycle

Definition: An Hamilton cycle is a cycle that traverses all nodes exactly once.

Proposition: It is NP-complete to decide whether or not a graph has an Hamilton cycle.

The Hamilton cycle problem is related to the travelling salesman problem, which asks for the shortest cycle that visits all nodes of a complete graph