Discrete Mathematics

Graphs

H. Turgut Uyar   Ayşegül Gençata Yayımlı   Emre Harmancı

2001-2016

License

© 2001-2016 T. Uyar, A. Yayımlı, E. Harmancı

You are free to:

- Share – copy and redistribute the material in any medium or format
- Adapt – remix, transform, and build upon the material

Under the following terms:

- Attribution – You must give appropriate credit, provide a link to the license, and indicate if changes were made.
- NonCommercial – You may not use the material for commercial purposes.
- ShareAlike – If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

For more information:
`https://creativecommons.org/licenses/by-nc-sa/4.0/`

Read the full license:
`https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode`

Topics

Graphs

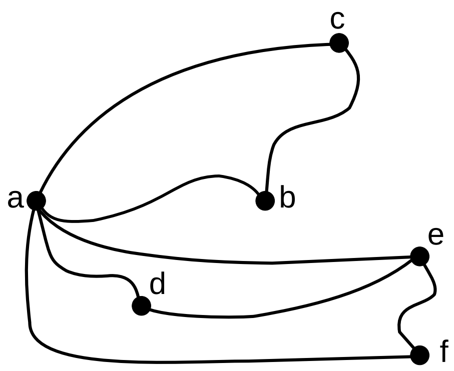
Definition

graph: $G = (V, E)$

- V: node (or *vertex*) set
- $E \subseteq V \times V$: edge set

- $e = (v_1, v_2) \in E$:
- v_1 and v_2 are *endnodes* of e
- e is *incident* to v_1 and v_2
- v_1 and v_2 are *adjacent*

Graph Example



$$
\begin{aligned}
V &= \{a, b, c, d, e, f\} \\
E &= \{(a, b), (a, c), \\
&\quad (a, d), (a, e), \\
&\quad (a, f), (b, c), \\
&\quad (d, e), (e, f)\}
\end{aligned}
$$

Directed Graphs

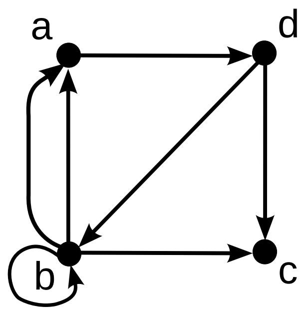
Definition

directed graph (*digraph*): $D = (V, A)$

- V: node set
- $A \subseteq V \times V$: arc set

- $a = (v_1, v_2) \in A$:
- v_1: *origin* node of a
- v_2: *terminating* node of a

Directed Graph Example



Weighted Graphs

- weighted graph: labels assigned to edges

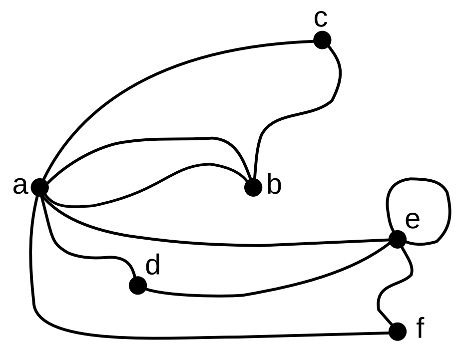
- weight
- length, distance
- cost, delay
- probability
- ...

Multigraphs

- parallel edges: edges between same node pair
- loop: edge starting and ending in same node

- plain graph: no loops, no parallel edges
- multigraph: a graph which is not plain

Multigraph Example



- parallel edges: (a, b)
- loop: (e, e)

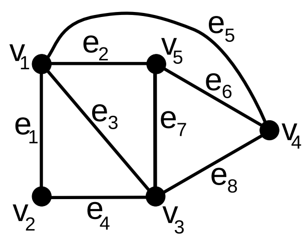
Subgraph

Definition
$G' = (V', E')$ is a subgraph of $G = (V, E)$:
$V' \subseteq V \wedge\ E' \subseteq E \wedge\ \forall (v_1, v_2) \in E'\ [v_1, v_2 \in V']$

Incidence Matrix

- rows: nodes, columns: edges
- 1 if edge incident on node, 0 otherwise

example

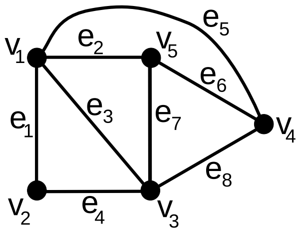


	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
v_1	1	1	1	0	1	0	0	0
v_2	1	0	0	1	0	0	0	0
v_3	0	0	1	1	0	0	1	1
v_4	0	0	0	0	1	1	0	1
v_5	0	1	0	0	0	1	1	0

Adjacency Matrix

- rows: nodes, columns: nodes
- 1 if nodes are adjacent, 0 otherwise

example

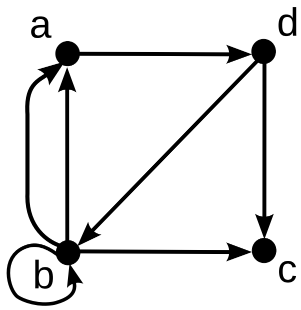


	v_1	v_2	v_3	v_4	v_5
v_1	0	1	1	1	1
v_2	1	0	1	0	0
v_3	1	1	0	1	1
v_4	1	0	1	0	1
v_5	1	0	1	1	0

Adjacency Matrix

- multigraph: number of edges between nodes

example



	a	b	c	d
a	0	0	0	1
b	2	1	1	0
c	0	0	0	0
d	0	1	1	0

Adjacency Matrix

- weighted graph: weight of edge

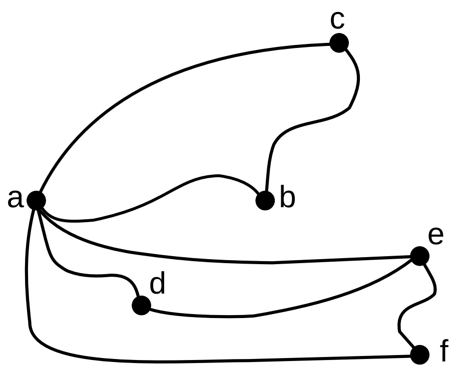
Degree

- degree of node: number of incident edges

Theorem
d_v: degree of node v

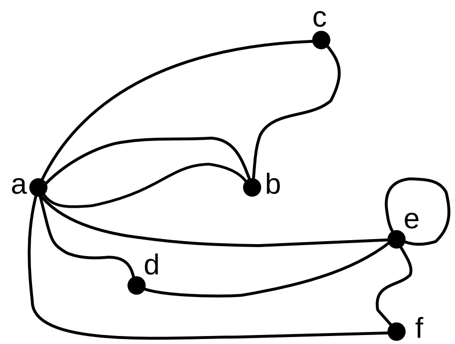
$$|E| = \frac{\sum_{v \in V} d_v}{2}$$

Degree Example



$$\begin{array}{rcr} d_a & = & 5 \\ d_b & = & 2 \\ d_c & = & 2 \\ d_d & = & 2 \\ d_e & = & 3 \\ d_f & = & 2 \\ \hline & & 16 \\ |E| & = & 8 \end{array}$$

Multigraph Degree Example



$$\begin{array}{rcr} d_a & = & 6 \\ d_b & = & 3 \\ d_c & = & 2 \\ d_d & = & 2 \\ d_e & = & 5 \\ d_f & = & 2 \\ \hline & & 20 \\ |E| & = & 10 \end{array}$$

Degree in Directed Graphs

- *in-degree*: $d_v{}^i$
- *out-degree*: $d_v{}^o$

- node with in-degree 0: *source*
- node with out-degree 0: *sink*

- $\sum_{v \in V} d_v{}^i = \sum_{v \in V} d_v{}^o = |A|$

Degree

Theorem

In an undirected graph, there is an even number of nodes which have an odd degree.

Proof.

- t_i: number of nodes of degree i
  $2|E| = \sum_{v \in V} d_v = 1t_1 + 2t_2 + 3t_3 + 4t_4 + 5t_5 + \dots$
  $2|E| - 2t_2 - 4t_4 - \dots = t_1 + t_3 + t_5 + \dots + 2t_3 + 4t_5 + \dots$
  $2|E| - 2t_2 - 4t_4 - \dots - 2t_3 - 4t_5 - \dots = t_1 + t_3 + t_5 + \dots$
- left-hand side even $\Rightarrow$ right-hand side even

$\square$

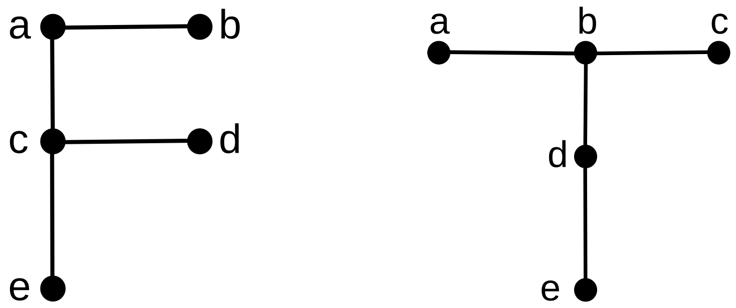
Isomorphism

Definition

$G = (V, E)$ and $G^\star = (V^\star, E^\star)$ are isomorphic:
$\exists f : V \rightarrow V^\star\ [(u, v) \in E \Rightarrow (f(u), f(v)) \in E^\star] \wedge f$ is bijective

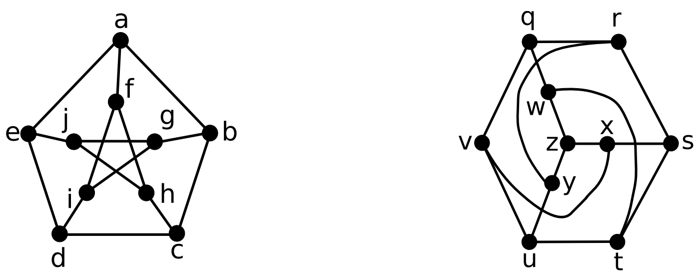
- G and $G^\star$ can be drawn the same way

Isomorphism Example



- $f = (a \mapsto d, b \mapsto e, c \mapsto b, d \mapsto c, e \mapsto a)$

Petersen Graph



- $f = (a \mapsto q, b \mapsto v, c \mapsto u, d \mapsto y, e \mapsto r,$
  $f \mapsto w, g \mapsto x, h \mapsto t, i \mapsto z, j \mapsto s)$

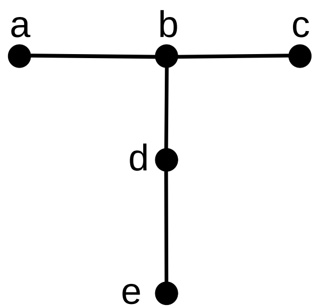
Homeomorphism

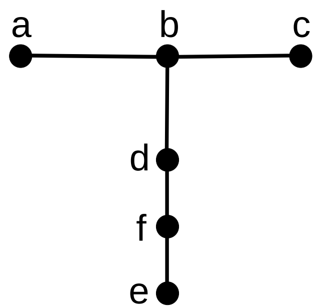
Definition

$G = (V, E)$ and $G^\star = (V^\star, E^\star)$ are homeomorphic:

- G and $G^\star$ isomorphic, except that
- some edges in $E^\star$ are divided with additional nodes

Homeomorphism Example

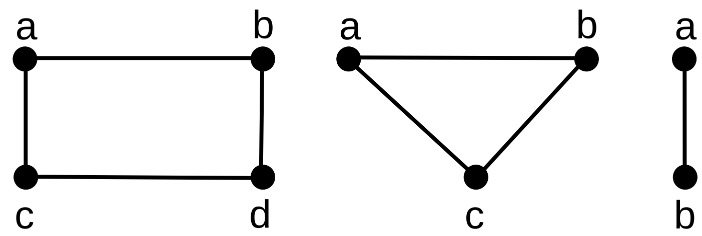




Regular Graphs

- regular graph: all nodes have the same degree
- n-regular: all nodes have degree n

examples

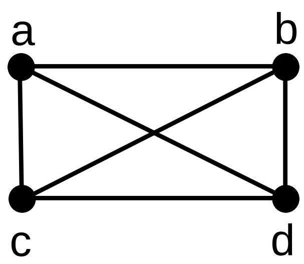


Completely Connected Graphs

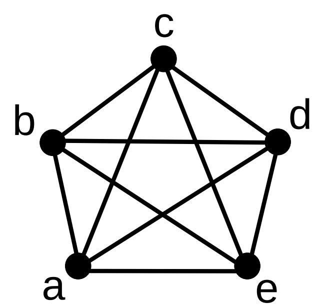
- $G = (V, E)$ is completely connected:
  $\forall v_1, v_2 \in V \; (v_1, v_2) \in E$
- every pair of nodes are adjacent
- K_n: completely connected graph with n nodes

Completely Connected Graph Examples

K_4



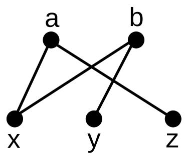
K_5



Bipartite Graphs

- $G = (V, E)$ is bipartite: $V_1 \cup V_2 = V$, $V_1 \cap V_2 = \emptyset$
  $\forall (v_1, v_2) \in E \; [v_1 \in V_1 \wedge v_2 \in V_2]$

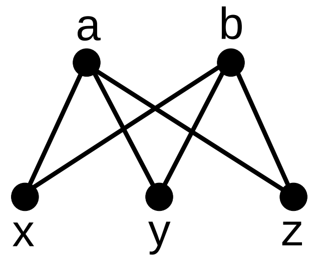
example



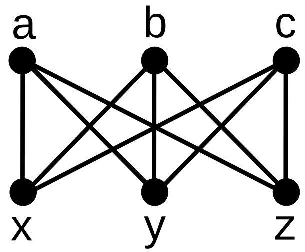
- *complete bipartite*: $\forall v_1 \in V_1 \; \forall v_2 \in V_2 \; (v_1, v_2) \in E$
- $K_{m,n}$: $|V_1| = m$, $|V_2| = n$

Complete Bipartite Graph Examples

$K_{2,3}$



$K_{3,3}$



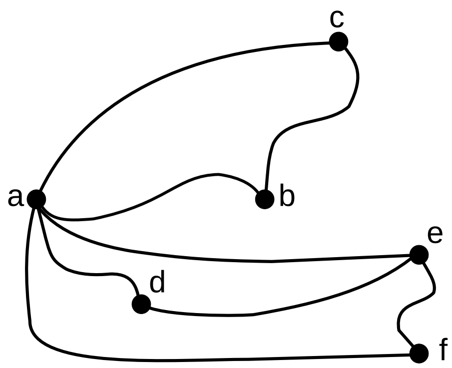
Walk

- walk: sequence of nodes and edges
  from a starting node (v_0) to an ending node (v_n)

$$v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} v_2 \xrightarrow{e_3} v_3 \longrightarrow \cdots \longrightarrow v_{n-1} \xrightarrow{e_n} v_n$$

  where $e_i = (v_{i-1}, v_i)$
- no need to write the edges if not weighted
- length: number of edges
- $v_0 = v_n$: closed

Walk Example



$$c \xrightarrow{(c,b)} b \xrightarrow{(b,a)} a \xrightarrow{(a,d)} d \xrightarrow{(d,e)} e \xrightarrow{(e,f)} f \xrightarrow{(f,a)} a \xrightarrow{(a,b)} b$$

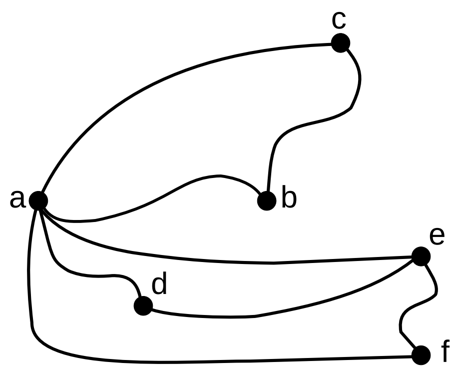
$$c \rightarrow b \rightarrow a \rightarrow d \rightarrow e \rightarrow f \rightarrow a \rightarrow b$$

length: 7

Trail

- trail: edges not repeated
- circuit: closed trail
- spanning trail: covers all edges

Trail Example



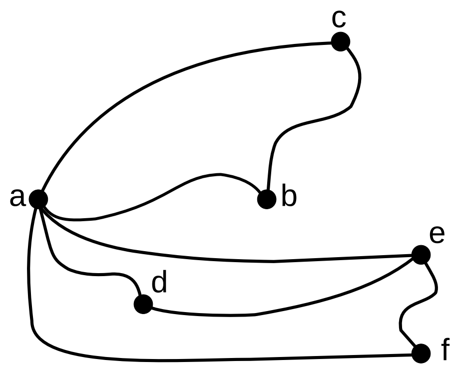
$c \xrightarrow{(c,b)} b \xrightarrow{(b,a)} a \xrightarrow{(a,e)} e \xrightarrow{(e,d)} d \xrightarrow{(d,a)} a \xrightarrow{(a,f)} f$

$c \rightarrow b \rightarrow a \rightarrow e \rightarrow d \rightarrow a \rightarrow f$

Path

- path: nodes not repeated
- cycle: closed path
- spanning path: visits all nodes

Path Example



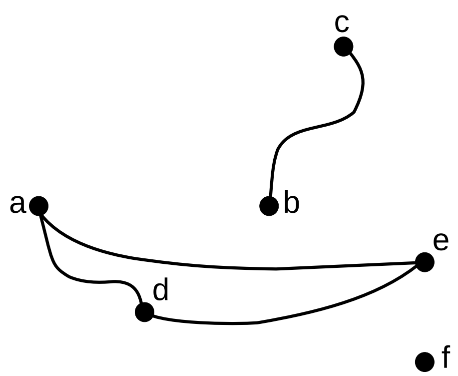
$c \xrightarrow{(c,b)} b \xrightarrow{(b,a)} a \xrightarrow{(a,d)} d \xrightarrow{(d,e)} e \xrightarrow{(e,f)} f$

$c \rightarrow b \rightarrow a \rightarrow d \rightarrow e \rightarrow f$

Connected Graphs

- connected: a path between every pair of nodes
- a disconnected graph can be divided into connected components

Connected Components Example

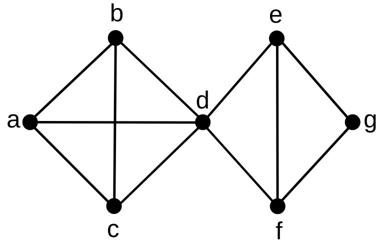


- disconnected: no path between a and c
- connected components: a, d, e b, c f

Distance

- distance between v_i and v_j: length of shortest path between v_i and v_j
- diameter of graph: largest distance in graph

Distance Example

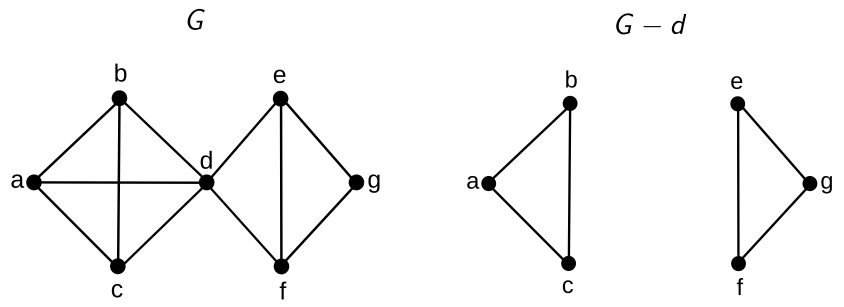


- distance between a and e: 2
- diameter: 3

Cut-Points

- $G - v$: delete v and all its incident edges from G
- v is a cut-point for G:
  G is connected but $G - v$ is not

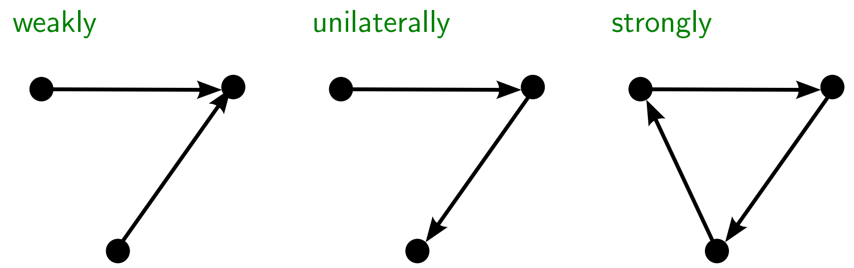
Cut-Point Example



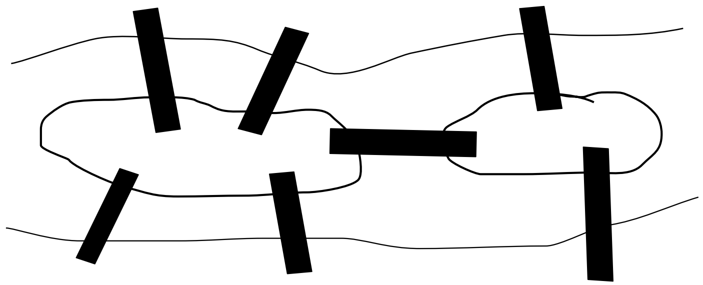
Directed Walks

- ignoring directions on arcs: *semi-walk*, *semi-trail*, *semi-path*
- if between every pair of nodes there is:
- a semi-path: weakly connected
- a path from one to the other: unilaterally connected
- a path: strongly connected

Directed Graph Examples

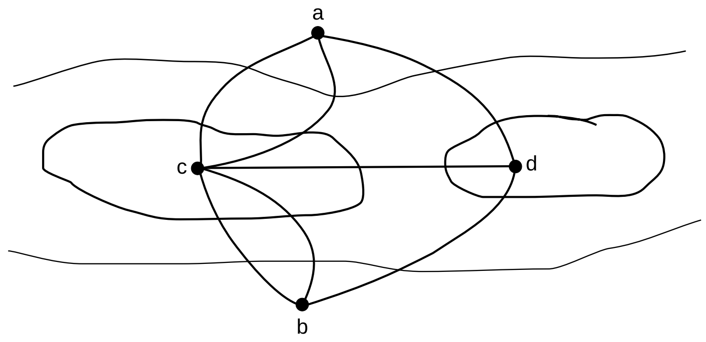


Bridges of Königsberg



- cross each bridge exactly once
  and return to the starting point

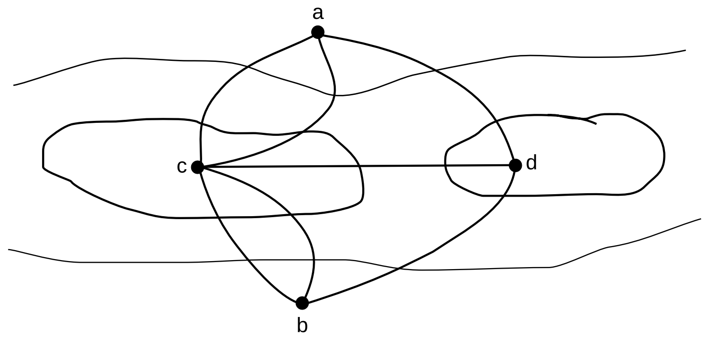
Graphs



Traversable Graphs

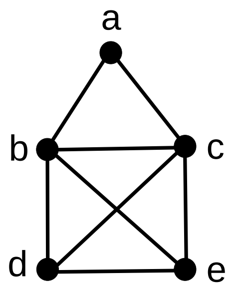
- G is traversable: G contains a spanning trail
- a node with an odd degree must be either the starting node or the ending node of the trail
- all nodes except the starting node and the ending node must have even degrees

Bridges of Königsberg



- all nodes have odd degrees: not traversable

Traversable Graph Example



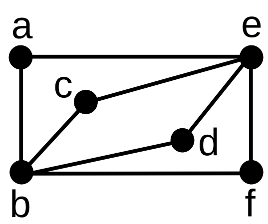
- a, b, c: even
- d, e: odd
- start from d, end at e:
  $d \to b \to a \to c \to e$
  $\to d \to c \to b \to e$

Euler Graphs

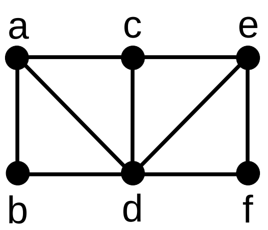
- Euler graph: contains closed spanning trail
- G is an Euler graph $\Leftrightarrow$ all nodes in G have even degrees

Euler Graph Examples

Euler



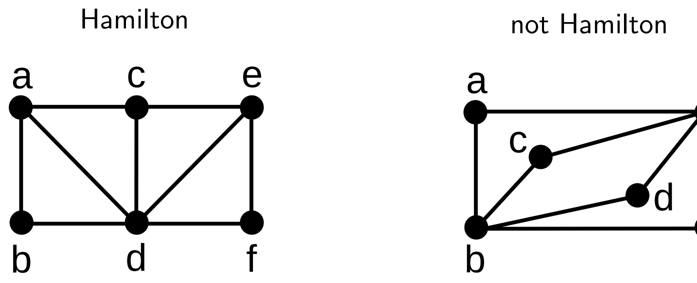
not Euler



Hamilton Graphs

- Hamilton graph: contains a closed spanning path

Hamilton Graph Examples



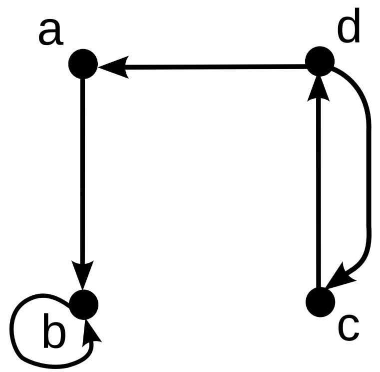
Connectivity Matrix

- A: adjacency matrix of $G = (V, E)$
- A^k_{ij}: number of walks of length k between v_i and v_j
- maximum distance between two nodes: $|V| - 1$
- connectivity matrix:
  $C = A^1 + A^2 + A^3 + \cdots + A^{|V|-1}$
- connected: all elements of C are non-zero

Warshall's Algorithm

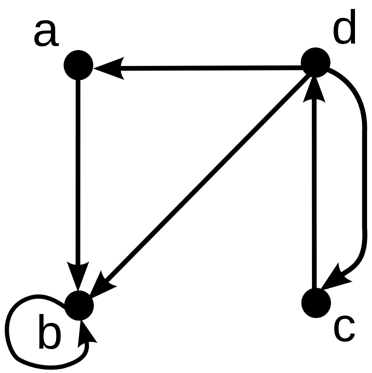
- very expensive to compute the connectivity matrix
- easier to find whether there is a walk between two nodes rather than finding the number of walks
- for each node:
- from all nodes which can reach the current node (rows that contain 1 in current column)
- to all nodes which can be reached from the current node (columns that contain 1 in current row)

Warshall's Algorithm Example



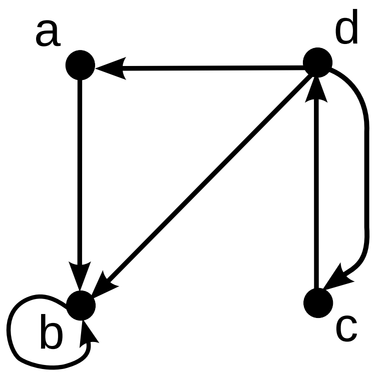
	a	b	c	d
a	0	1	0	0
b	0	1	0	0
c	0	0	0	1
d	1	0	1	0

Warshall's Algorithm Example



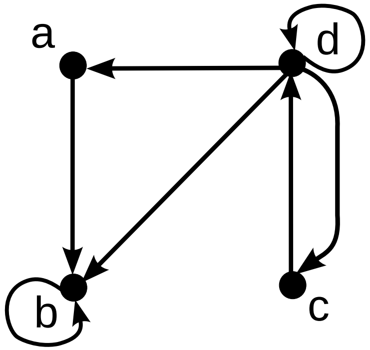
	a	b	c	d
a	0	1	0	0
b	0	1	0	0
c	0	0	0	1
d	1	1	1	0

Warshall's Algorithm Example



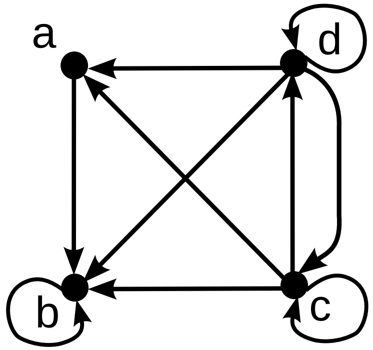
	a	b	c	d
a	0	1	0	0
b	0	1	0	0
c	0	0	0	1
d	1	1	1	0

Warshall's Algorithm Example



	a	b	c	d
a	0	1	0	0
b	0	1	0	0
c	0	0	0	1
d	1	1	1	1

Warshall's Algorithm Example



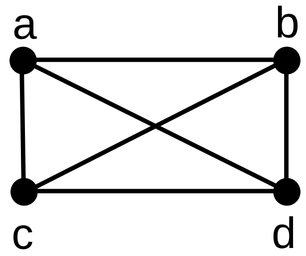
	a	b	c	d
a	0	1	0	0
b	0	1	0	0
c	1	1	1	1
d	1	1	1	1

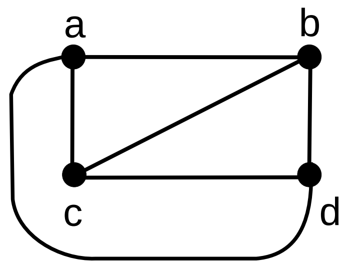
Planar Graphs

Definition
G is planar:
G can be drawn on a plane without intersecting its edges

- a map of G: a planar drawing of G

Planar Graph Example





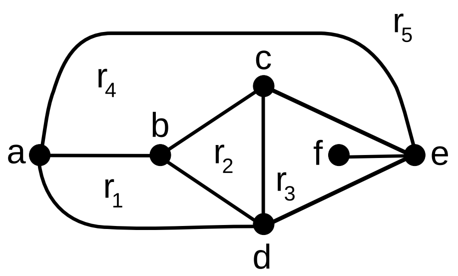
Regions

- map divides plane into regions
- degree of region: length of closed trail that surrounds region

Theorem
d_{r_i}: degree of region r_i

$$|E| = \frac{\sum_i d_{r_i}}{2}$$

Region Example



$$\begin{array}{rcr}
d_{r_1} & = & 3 \\
d_{r_2} & = & 3 \\
d_{r_3} & = & 5 \\
d_{r_4} & = & 4 \\
d_{r_5} & = & 3 \\
\hline
 & = & 18 \\
|E| & = & 9
\end{array}$$

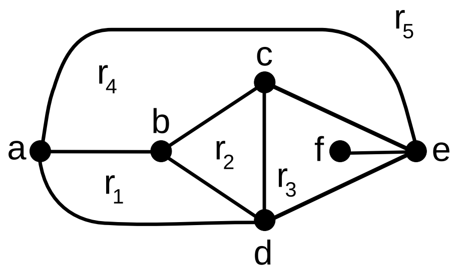
Euler's Formula

Theorem (Euler's Formula)

$G = (V, E)$: planar, connected graph
R: set of regions in a map of G

$$|V| - |E| + |R| = 2$$

Euler's Formula Example



- $|V| = 6$, $|E| = 9$, $|R| = 5$

Planar Graph Theorems

Theorem

$G = (V, E)$: connected, planar graph where $|V| \geq 3$
$|E| \leq 3|V| - 6$

Proof.

- sum of region degrees: $2|E|$
- degree of a region ≥ 3
  $\Rightarrow 2|E| \geq 3|R| \Rightarrow |R| \leq \frac{2}{3}|E|$
- $|V| - |E| + |R| = 2$
  $\Rightarrow |V| - |E| + \frac{2}{3}|E| \geq 2 \Rightarrow |V| - \frac{1}{3}|E| \geq 2$
  $\Rightarrow 3|V| - |E| \geq 6 \Rightarrow |E| \leq 3|V| - 6$

$\square$

Planar Graph Theorems

Theorem

$G = (V, E)$: connected, planar graph where $|V| \geq 3$:
$\exists v \in V\ [d_v \leq 5]$

Proof.

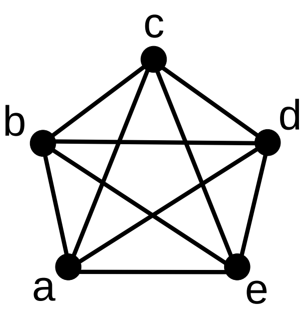
- assume: $\forall v \in V\ [d_v \geq 6]$
  $\Rightarrow 2|E| \geq 6|V|$
  $\Rightarrow |E| \geq 3|V|$
  $\Rightarrow |E| > 3|V| - 6$

$\square$

Nonplanar Graphs

Theorem

K_5 is not planar.



Proof.

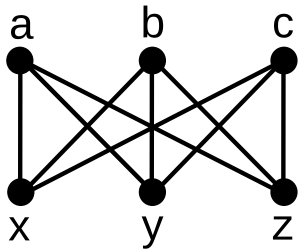
- $|V| = 5$
- $3|V| - 6 = 3 \cdot 5 - 6 = 9$
- $|E| \leq 9$ should hold
- but $|E| = 10$

$\square$

Nonplanar Graphs

Theorem

$K_{3,3}$ is not planar.



Proof.

- $|V| = 6, |E| = 9$
- if planar then $|R| = 5$
- degree of a region ≥ 4
  $\Rightarrow \sum_{r \in R} d_r \geq 20$
- $|E| \geq 10$ should hold
- but $|E| = 9$

$\square$

Kuratowski's Theorem

Theorem

G contains a subgraph homeomorphic to K_5 or $K_{3,3}$.

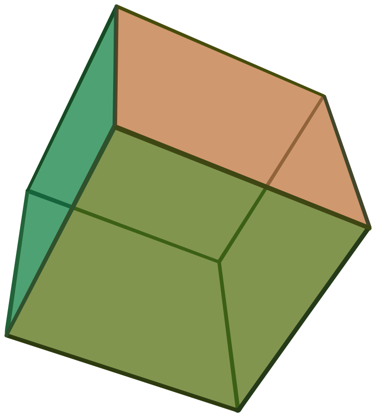
$\Leftrightarrow$

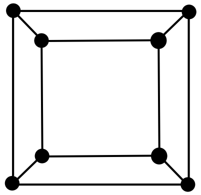
G is not planar.

Platonic Solids

- *regular polyhedron*: a 3-dimensional solid where faces are identical regular polygons
- projection of a regular polyhedron onto the plane: a planar graph
- corners: nodes
- sides: edges
- faces: regions

Platonic Solid Example





Platonic Solids

- $|V|$: number of corners (nodes)
- $|E|$: number of sides (edges)
- $|R|$: number of faces (regions)
- n: number of faces meeting at a corner (node degree)
- m: number of sides of a face (region degree)

- $m, n \geq 3$
- $2|E| = n \cdot |V|$
- $2|E| = m \cdot |R|$

Platonic Solids

- from Euler's formula:

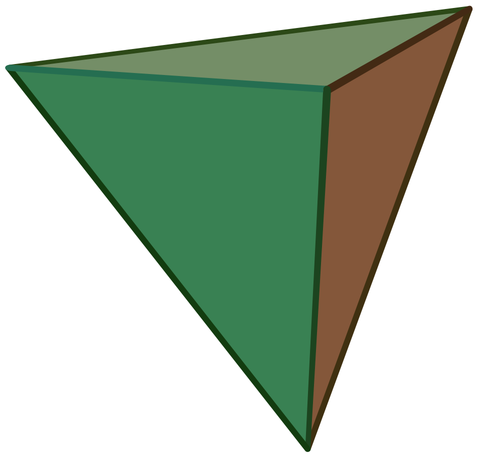
$$2 = |V| - |E| + |R| = \frac{2|E|}{n} - |E| + \frac{2|E|}{m} = |E| \left(\frac{2m - mn + 2n}{mn} \right) > 0$$

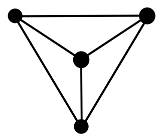
- $|E|, m, n > 0$:

$$2m - mn + 2n > 0 \Rightarrow mn - 2m - 2n < 0$$
$$\Rightarrow mn - 2m - 2n + 4 < 4 \Rightarrow (m-2)(n-2) < 4$$

- only 5 solutions

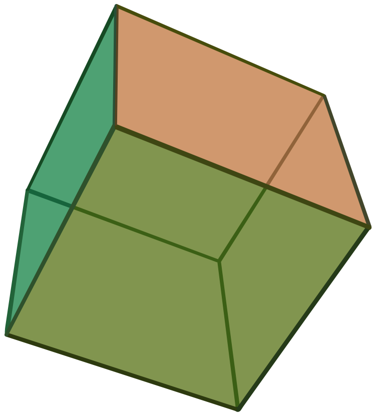
Tetrahedron

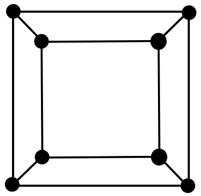




$m = 3, n = 3$

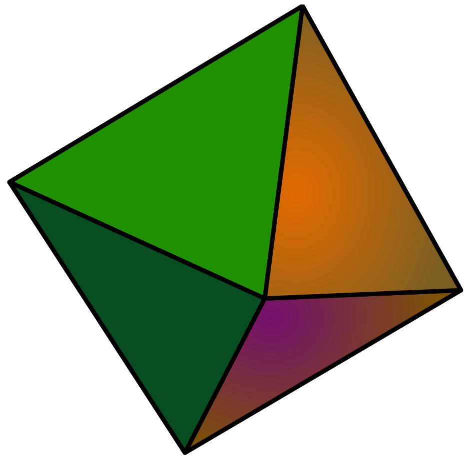
Hexahedron

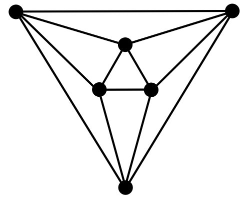




$m = 4, n = 3$

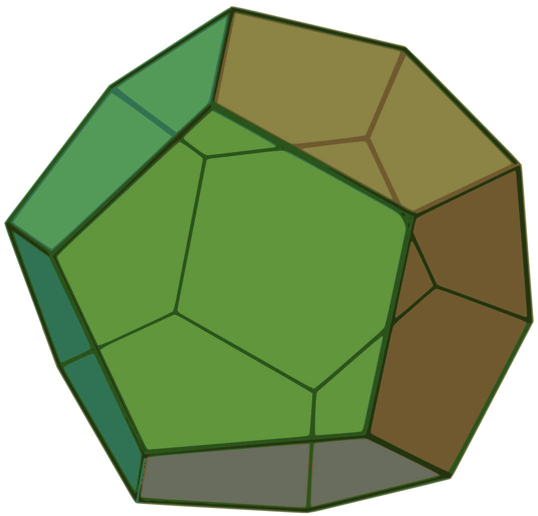
Octahedron

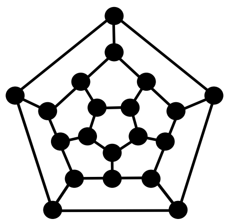




$m = 3, n = 4$

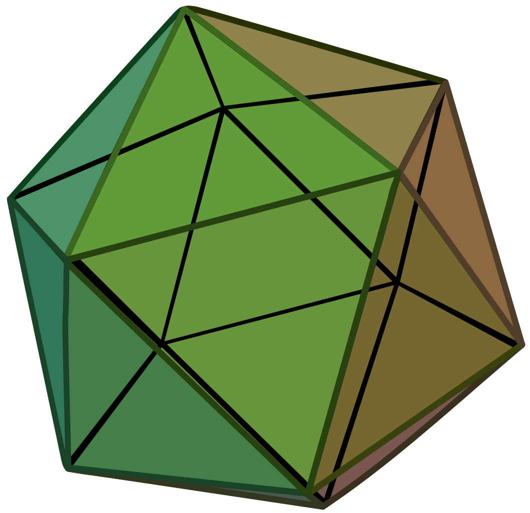
Dodecahedron





$m = 5, n = 3$

Icosahedron

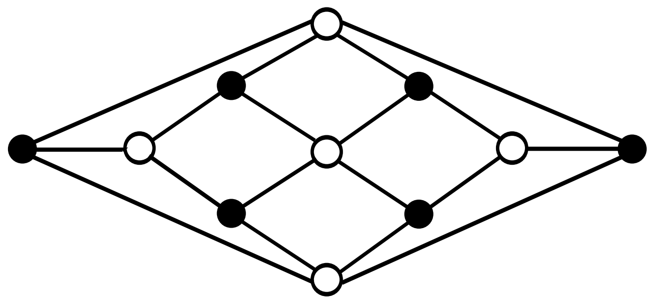


$m = 3, n = 5$

Graph Coloring

- $G = (V, E)$, C: set of colors
- proper coloring of G: find an $f : V \to C$, such that $\forall(v_i, v_j) \in E\ [f(v_i) \neq f(v_j)]$
- chromatic number of G: $\chi(G)$ minimum $|C|$
- finding $\chi(G)$ is a very difficult problem
- $\chi(K_n) = n$

Chromatic Number Example



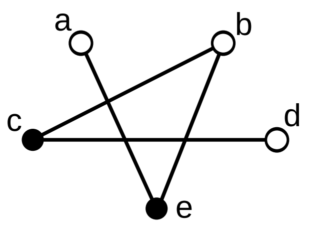
- Herschel graph: $\chi(G) = 2$

Graph Coloring Example

- a company produces chemical compounds
- some compounds cannot be stored together
- such compounds must be placed in separate storage areas
- store compounds using minimum number of storage areas

Graph Coloring Example

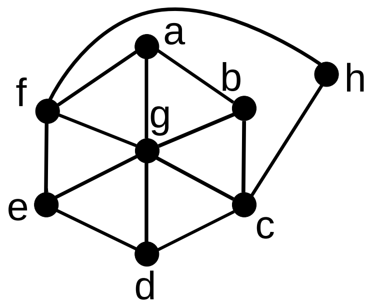
- every compound is a node
- two compounds that cannot be stored together are adjacent



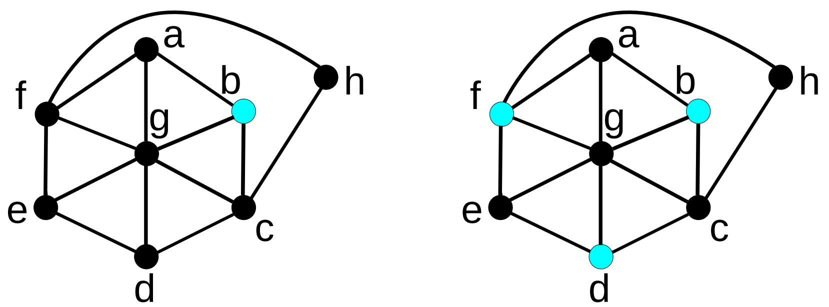
Graph Coloring Solution

- pick a node and assign a color
- assign same color to all nodes with no conflict
- pick an uncolored node and assign a second color
- assign same color to all uncolored nodes with no conflict
- pick an uncolored node and assign a third color
- ...

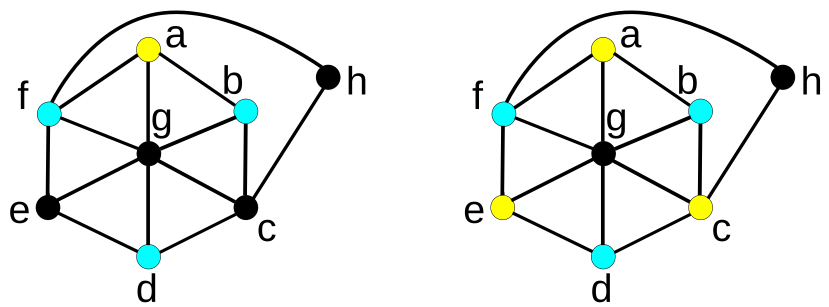
Graph Coloring Example



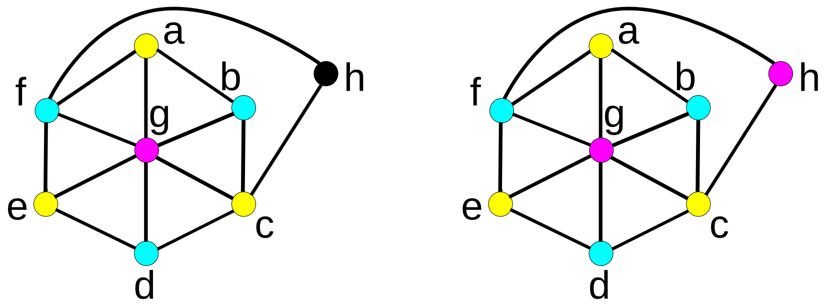
Graph Coloring Example



Graph Coloring Example



Graph Coloring Example



Heuristic Solutions

- heuristic solution: based on intuition
- greedy solution: doesn't look ahead
- doesn't produce optimal results

Graph Coloring Example: Sudoku

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

- every cell is a node
- cells of the same row are adjacent
- cells of the same column are adjacent
- cells of the same 3×3 block are adjacent
- every number is a color
- problem: properly color a graph that is partially colored

Region Coloring

- coloring a map by assigning different colors to adjacent regions

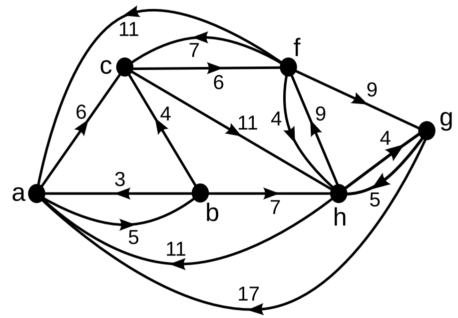
Theorem (Four Color Theorem)

The regions in a map can be colored using four colors.

Shortest Path

- finding shortest paths from a starting node to all other nodes: Dijkstra's algorithm

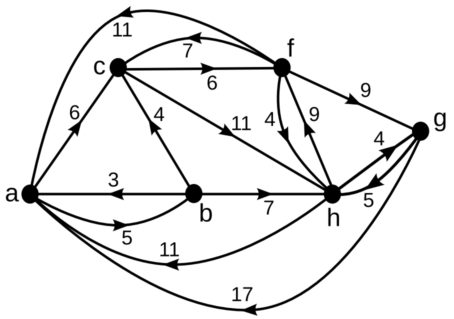
Dijkstra's Algorithm Example



- starting node: c

a	$(\infty, -)$
b	$(\infty, -)$
c	$(0, -)$
f	$(\infty, -)$
g	$(\infty, -)$
h	$(\infty, -)$

Dijkstra's Algorithm Example

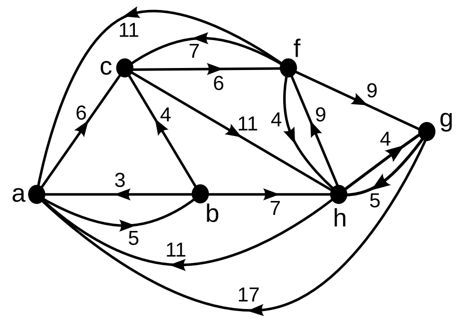


- from c: base distance=0
- $c \to f : 6, 6 < \infty$
- $c \to h : 11, 11 < \infty$

a	$(\infty, -)$	
b	$(\infty, -)$	
c	$(0, -)$	$\surd$
f	$(6, cf)$	
g	$(\infty, -)$	
h	$(11, ch)$	

- closest node: f

Dijkstra's Algorithm Example

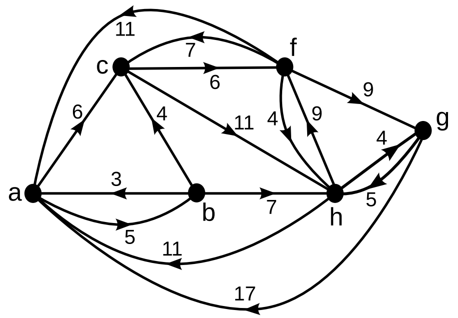


- from f: base distance=6
- $f \to a : 6 + 11, 17 < \infty$
- $f \to g : 6 + 9, 15 < \infty$
- $f \to h : 6 + 4, 10 < 11$

a	$(17, cfa)$	
b	$(\infty, -)$	
c	$(0, -)$	$\surd$
f	$(6, cf)$	$\surd$
g	$(15, cfg)$	
h	$(10, cfh)$	

- closest node: h

Dijkstra's Algorithm Example

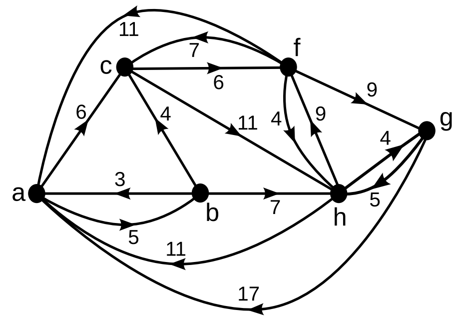


- from h: base distance=10
- $h \rightarrow a : 10 + 11, 21 \not< 17$
- $h \rightarrow g : 10 + 4, 14 < 15$

a	$(17, cfa)$	
b	$(\infty, -)$	
c	$(0, -)$	$\surd$
f	$(6, cf)$	$\surd$
g	$(14, cfhg)$	
h	$(10, cfh)$	$\surd$

- closest node: g

Dijkstra's Algorithm Example

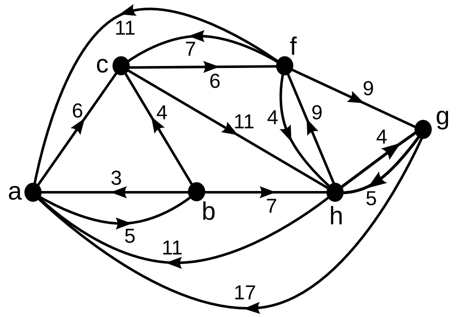


- from g: base distance=14
- $g \rightarrow a : 14 + 17, 31 \not< 17$

a	$(17, cfa)$	
b	$(\infty, -)$	
c	$(0, -)$	$\surd$
f	$(6, cf)$	$\surd$
g	$(14, cfhg)$	$\surd$
h	$(10, cfh)$	$\surd$

- closest node: a

Dijkstra's Algorithm Example



- from a: base distance=17
- $a \rightarrow b : 17 + 5, 22 < \infty$

a	$(17, cfa)$	$\surd$
b	$(22, cfab)$	
c	$(0, -)$	$\surd$
f	$(6, cf)$	$\surd$
g	$(14, cfhg)$	$\surd$
h	$(10, cfh)$	$\surd$

- last node: b

Traveling Salesperson Problem

- start from a home town
- visit every city exactly once
- return to the home town
- minimum total distance

- find Hamiltonian cycle
- very difficult problem

TSP Solution

- heuristic: nearest-neighbor

Searching Graphs

- searching nodes of graph $G = (V, E)$ starting from node v_1
- depth-first
- breadth-first

Depth-First Search

1. $v \leftarrow v_1$, $T = \emptyset$, $D = \{v_1\}$
2. find smallest i in $2 \leq i \leq |V|$ such that $(v, v_i) \in E$ and $v_i \notin D$
   - if no such i: go to step 3
   - if found: $T = T \cup \{(v, v_i)\}$, $D = D \cup \{v_i\}$, $v \leftarrow v_i$, go to step 2
3. if $v = v_1$: result is T
4. if $v \neq v_1$: $v \leftarrow \mathit{backtrack}(v)$, go to step 2

Breadth-First Search

1. $T = \emptyset$, $D = \{v_1\}$, $Q = (v_1)$
2. if Q empty: result is T
3. if Q not empty: $v \leftarrow \mathit{front}(Q)$, $Q \leftarrow Q - v$
   for $2 \leq i \leq |V|$ check edges $(v, v_i) \in E$:
   - if $v_i \notin D$: $Q = Q + v_i$, $T = T \cup \{(v, v_i)\}$, $D = D \cup \{v_i\}$
   - go to step 3

References

Required Reading: Grimaldi

- Chapter 11: An Introduction to Graph Theory
- Chapter 7: Relations: The Second Time Around
  - 7.2. Computer Recognition: Zero-One Matrices and Directed Graphs
- Chapter 13: Optimization and Matching
  - 13.1. Dijkstra's Shortest Path Algorithm