Graphs

Lecture 12

Story so far

Story so far

- Simple graph G = (V,E): undirected, no self-loops, no multi-edges: a symmetric 0-1 adjacency matrix with an all-zero diagonal

Story so far

- Simple graph G = (V,E): undirected, no self-loops, no multi-edges: a symmetric 0-1 adjacency matrix with an all-zero diagonal
- e.g.: clique K_n, cycle C_n, bipartite graphs

Story so far

- Simple graph G = (V,E): undirected, no self-loops, no multi-edges: a symmetric 0-1 adjacency matrix with an all-zero diagonal
- e.g.: clique K_n, cycle C_n, bipartite graphs
- Graph isomorphism: a bijection f between V_1 and V_2 s.t. $\{u,v\} \in E_1$ iff $\{f(u),f(v)\} \in E_2$

Story so far

- Simple graph G = (V,E): undirected, no self-loops, no multi-edges: a symmetric 0-1 adjacency matrix with an all-zero diagonal
- e.g.: clique K_n, cycle C_n, bipartite graphs
- Graph isomorphism: a bijection f between V_1 and V_2 s.t. $\{u,v\} \in E_1$ iff $\{f(u),f(v)\} \in E_2$
  - e.g. C_4 and $K_{2,2}$. $K_{1,2}$ and P_3 (path graph)

Story so far

- Simple graph G = (V,E): undirected, no self-loops, no multi-edges: a symmetric 0-1 adjacency matrix with an all-zero diagonal
- e.g.: clique K_n, cycle C_n, bipartite graphs
- Graph isomorphism: a bijection f between V_1 and V_2 s.t. $\{u,v\} \in E_1$ iff $\{f(u),f(v)\} \in E_2$
  - e.g. C_4 and $K_{2,2}$. $K_{1,2}$ and P_3 (path graph)
- Subgraphs: remove zero or more vertices (and incident edged), and then zero or more additional edges

Story so far

- Simple graph G = (V,E): undirected, no self-loops, no multi-edges: a symmetric 0-1 adjacency matrix with an all-zero diagonal
- e.g.: clique K_n, cycle C_n, bipartite graphs
- Graph isomorphism: a bijection f between V_1 and V_2 s.t. $\{u,v\} \in E_1$ iff $\{f(u),f(v)\} \in E_2$
  - e.g. C_4 and $K_{2,2}$. $K_{1,2}$ and P_3 (path graph)
- Subgraphs: remove zero or more vertices (and incident edged), and then zero or more additional edges
  - Cyclic: has C_n as a subgraph for some n, $3 \le n \le |V|$. Else acyclic (a.k.a, tree, for simple graphs)

Story so far

- Simple graph G = (V,E): undirected, no self-loops, no multi-edges: a symmetric 0-1 adjacency matrix with an all-zero diagonal
- e.g.: clique K_n, cycle C_n, bipartite graphs
- Graph isomorphism: a bijection f between V_1 and V_2 s.t. $\{u,v\} \in E_1$ iff $\{f(u),f(v)\} \in E_2$
  - e.g. C_4 and $K_{2,2}$. $K_{1,2}$ and P_3 (path graph)
- Subgraphs: remove zero or more vertices (and incident edged), and then zero or more additional edges
  - Cyclic: has C_n as a subgraph for some n, $3 \le n \le |V|$. Else acyclic (a.k.a, tree, for simple graphs)
- Degree of a node: $\Sigma_{v \in V} \deg(v) = 2|E|$

Story so far

- Simple graph G = (V,E): undirected, no self-loops, no multi-edges: a symmetric 0-1 adjacency matrix with an all-zero diagonal
- e.g.: clique K_n, cycle C_n, bipartite graphs
- Graph isomorphism: a bijection f between V_1 and V_2 s.t. $\{u,v\} \in E_1$ iff $\{f(u),f(v)\} \in E_2$
  - e.g. C_4 and $K_{2,2}$. $K_{1,2}$ and P_3 (path graph)
- Subgraphs: remove zero or more vertices (and incident edged), and then zero or more additional edges
  - Cyclic: has C_n as a subgraph for some n, $3 \le n \le |V|$. Else acyclic (a.k.a, tree, for simple graphs)
- Degree of a node: $\Sigma_{v \in V} \deg(v) = 2|E|$
- u connected to v iff there is a path (equivalently, a walk) between u and v. Equivalence classes: Connected components.

Question

Question

- How many edges are there in the clique, K_6?

A. 30
B. 15
C. 36
D. 18
E. 25

Question

- How many edges are there in the clique, K_6?

A. 30
B. 15
C. 36
D. 18
E. 25

n(n-1)/2

Question

Question

- How many edges are there in the complete bi-partite graph, $K_{6,6}$?

A. 30
B. 15
C. 36
D. 18
E. 25

Question

- How many edges are there in the complete bi-partite graph, $K_{6,6}$?

A. 30
B. 15
C. 36
D. 18
E. 25

$n_1.n_2$

Question

Question

- Out of K_3, $K_{2,2}$, C_5, C_6, which ones are bi-partite?

A. All of them
B. $K_{2,2}$ & C_6 only
C. $K_{2,2}$, C_5 & C_6 only
D. C_5 & C_6
E. None of the above choices

Question

- Out of K_3, $K_{2,2}$, C_5, C_6, which ones are bi-partite?

A. All of them
B. $K_{2,2}$ & C_6 only
C. $K_{2,2}$, C_5 & C_6 only
D. C_5 & C_6
E. None of the above choices

Bi-partite graphs can't have an **odd-cycle** as a subgraph

Question

- Out of K_3, $K_{2,2}$, C_5, C_6, which ones are bi-partite?

A. All of them
B. $K_{2,2}$ & C_6 only
C. $K_{2,2}$, C_5 & C_6 only
D. C_5 & C_6
E. None of the above choices

Bi-partite graphs can't have an **odd-cycle** as a subgraph

In fact, converse also holds. Bi-partite iff no odd-cycle!

Question

Question

- A graph is said to be d-regular if all nodes have degree d. How many edges are there in a 3-regular graph with 8 nodes?

    A. Such a graph doesn't exist
    B. 24
    C. 48
    D. 12
    E. 32

Question

- A graph is said to be d-regular if all nodes have degree d. How many edges are there in a 3-regular graph with 8 nodes?

A. Such a graph doesn't exist
B. 24
C. 48
D. 12
E. 32

n·d/2 (unless n & d odd)

Another Example

Another Example

- The hypercube graph Q_n

Another Example

- The hypercube graph Q_n
  - Nodes: all n-bit strings. e.g., { 000,001,010,011,100,101,110,111}

Another Example

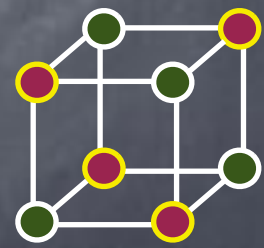
- The hypercube graph Q_n
  - Nodes: all n-bit strings. e.g., { 000,001,010,011,100,101,110,111}
  - Edges: x and y connected iff they differ in exactly one position

Another Example

- The hypercube graph Q_n
  - Nodes: all n-bit strings. e.g., { 000,001,010,011,100,101,110,111}
  - Edges: x and y connected iff they differ in exactly one position
    - i.e., x & y neighbors if toggling a single bit changes x to y

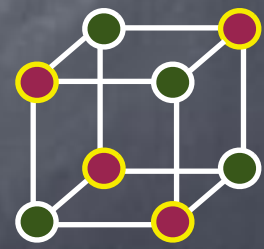
Another Example

- The hypercube graph Q_n
  - Nodes: all n-bit strings. e.g., { 000,001,010,011,100,101,110,111}
  - Edges: x and y connected iff they differ in exactly one position
    - i.e., x & y neighbors if toggling a single bit changes x to y
  - e.g. Q_3 can be drawn like a "cube"



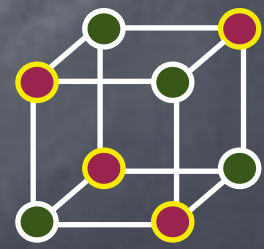
Another Example

- The hypercube graph Q_n
  - Nodes: all n-bit strings. e.g., { 000,001,010,011,100,101,110,111}
  - Edges: x and y connected iff they differ in exactly one position
    - i.e., x & y neighbors if toggling a single bit changes x to y
  - e.g. Q_3 can be drawn like a "cube"
- 2^n nodes, but "diameter" (longest shortest path) is only n



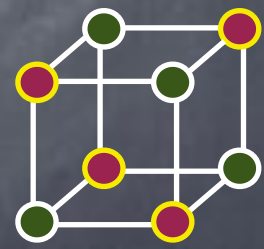
Another Example

- The hypercube graph Q_n
  - Nodes: all n-bit strings. e.g., { 000,001,010,011,100,101,110,111}
  - Edges: x and y connected iff they differ in exactly one position
    - i.e., x & y neighbors if toggling a single bit changes x to y
  - e.g. Q_3 can be drawn like a "cube"
- 2^n nodes, but "diameter" (longest shortest path) is only n
- Q_n is an n-regular bi-partite graph



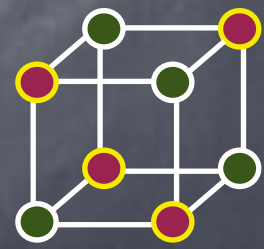
Another Example

- The hypercube graph Q_n
  - Nodes: all n-bit strings. e.g., { 000,001,010,011,100,101,110,111}
  - Edges: x and y connected iff they differ in exactly one position
    - i.e., x & y neighbors if toggling a single bit changes x to y
  - e.g. Q_3 can be drawn like a "cube"
- 2^n nodes, but "diameter" (longest shortest path) is only n
- Q_n is an n-regular bi-partite graph
  - The two parts: nodes labeled with strings which have even parity (even# 1s) and those labeled with strings of odd parity (odd# 1s)



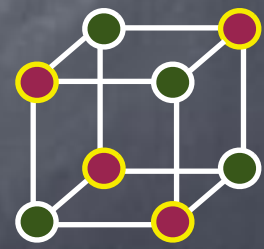
Another Example

- The hypercube graph Q_n
  - Nodes: all n-bit strings. e.g., { 000,001,010,011,100,101,110,111}
  - Edges: x and y connected iff they differ in exactly one position
    - i.e., x & y neighbors if toggling a single bit changes x to y
  - e.g. Q_3 can be drawn like a "cube"
- 2^n nodes, but "diameter" (longest shortest path) is only n
- Q_n is an n-regular bi-partite graph
  - The two parts: nodes labeled with strings which have even parity (even# 1s) and those labeled with strings of odd parity (odd# 1s)
- Q_{n-1} is a subgraph of Q_n



Another Example

- The hypercube graph Q_n
  - Nodes: all n-bit strings. e.g., { 000,001,010,011,100,101,110,111}
  - Edges: x and y connected iff they differ in exactly one position
    - i.e., x & y neighbors if toggling a single bit changes x to y
  - e.g. Q_3 can be drawn like a "cube"

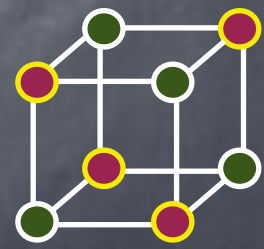


- 2^n nodes, but "diameter" (longest shortest path) is only n
- Q_n is an n-regular bi-partite graph
  - The two parts: nodes labeled with strings which have even parity (even# 1s) and those labeled with strings of odd parity (odd# 1s)

- Q_{n-1} is a subgraph of Q_n

Another Example

- The hypercube graph Q_n
  - Nodes: all n-bit strings. e.g., { 000,001,010,011,100,101,110,111}
  - Edges: x and y connected iff they differ in exactly one position
    - i.e., x & y neighbors if toggling a single bit changes x to y
  - e.g. Q_3 can be drawn like a "cube"

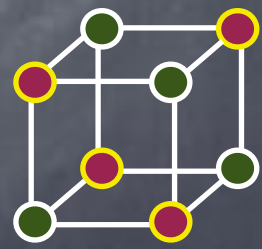


- 2^n nodes, but "diameter" (longest shortest path) is only n
- Q_n is an n-regular bi-partite graph
  - The two parts: nodes labeled with strings which have even parity (even# 1s) and those labeled with strings of odd parity (odd# 1s)

- Q_{n-1} is a subgraph of Q_n

Another Example

- The hypercube graph Q_n
  - Nodes: all n-bit strings. e.g., { 000,001,010,011,100,101,110,111}
  - Edges: x and y connected iff they differ in exactly one position
    - i.e., x & y neighbors if toggling a single bit changes x to y
  - e.g. Q_3 can be drawn like a "cube"
- 2^n nodes, but "diameter" (longest shortest path) is only n



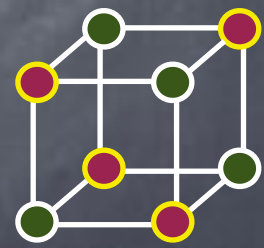
- Q_n is an n-regular bi-partite graph
  - The two parts: nodes labeled with strings which have <u>even parity</u> (even# 1s) and those labeled with strings of <u>odd parity</u> (odd# 1s)
- Q_{n-1} is a subgraph of Q_n

"$0Q_3$"

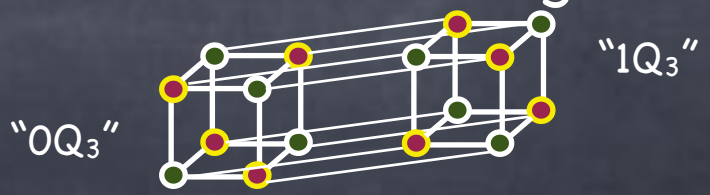
"$1Q_3$"

Another Example

- The hypercube graph Q_n
  - Nodes: all n-bit strings. e.g., { 000,001,010,011,100,101,110,111}
  - Edges: x and y connected iff they differ in exactly one position
    - i.e., x & y neighbors if toggling a single bit changes x to y
  - e.g. Q_3 can be drawn like a "cube"



- 2^n nodes, but "diameter" (longest shortest path) is only n
- Q_n is an n-regular bi-partite graph
  - The two parts: nodes labeled with strings which have <u>even parity</u> (even# 1s) and those labeled with strings of <u>odd parity</u> (odd# 1s)
- Q_{n-1} is a subgraph of Q_n



Question

Question

- In Q_5, what is the distance (length of a shortest path) between the nodes labeled 00100 and 10001?

A. 6
B. 5
C. 4
D. 3
E. 2

Question

- In Q_5, what is the distance (length of a shortest path) between the nodes labeled 00100 and 10001?

A. 6
B. 5
C. 4
D. 3
E. 2

"weight of $x \oplus y$"

Question

- In Q_5, what is the distance (length of a shortest path) between the nodes labeled 00100 and 10001?

  A. 6
  B. 5
  C. 4
  D. 3
  E. 2

"weight of $x \oplus y$"

- Many shortest paths

Question

Question

- In Q_5, how many shortest paths (of length 3) are there between the nodes labeled 00100 and 10001?

A. 32
B. 8
C. 6
D. 4
E. 3

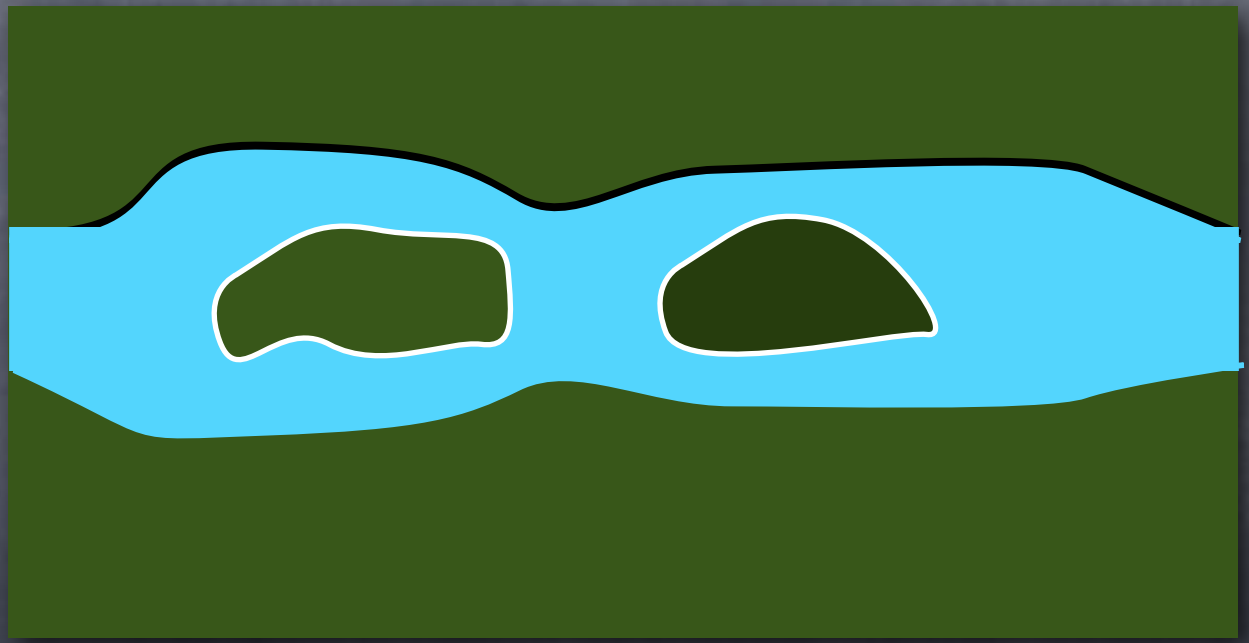
Question

- In Q_5, how many shortest paths (of length 3) are there between the nodes labeled 00100 and 10001?

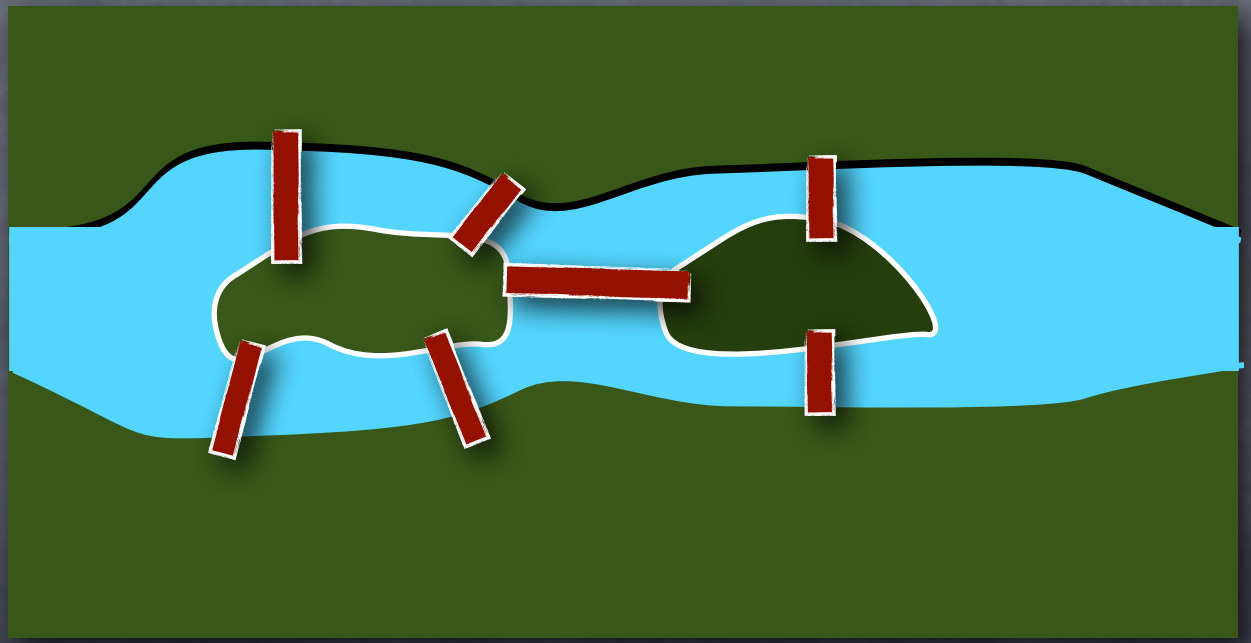
A. 32
B. 8
C. 6
D. 4
E. 3

3! ways to make 3 toggles

Bridges of Königsberg

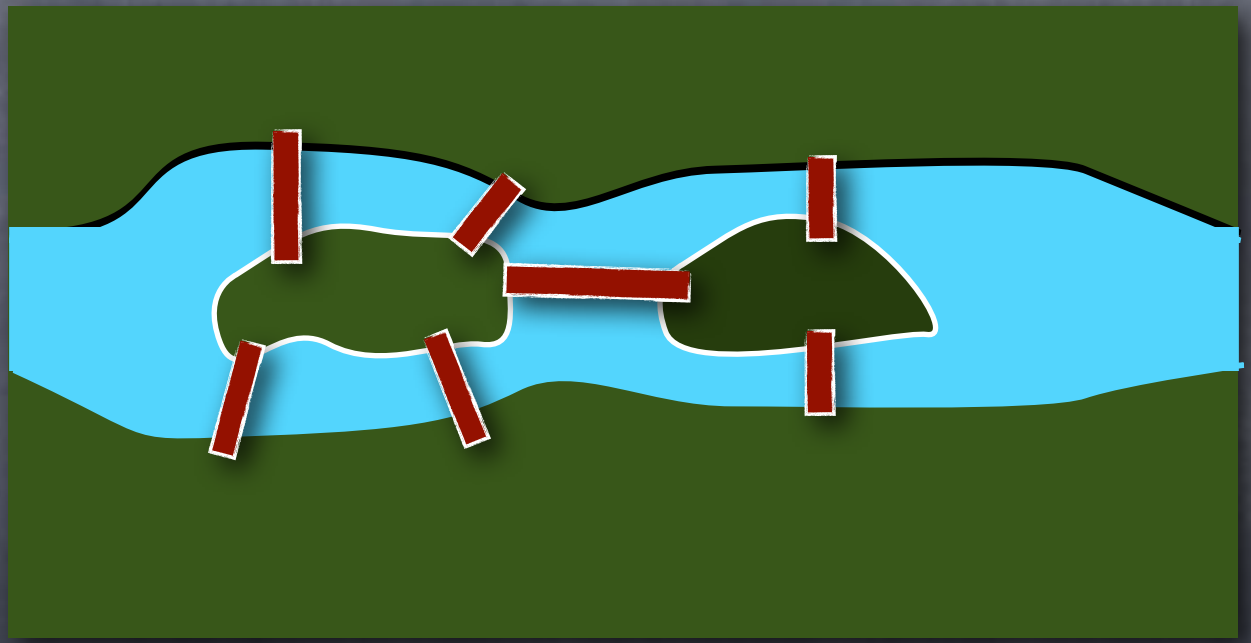


Bridges of Königsberg



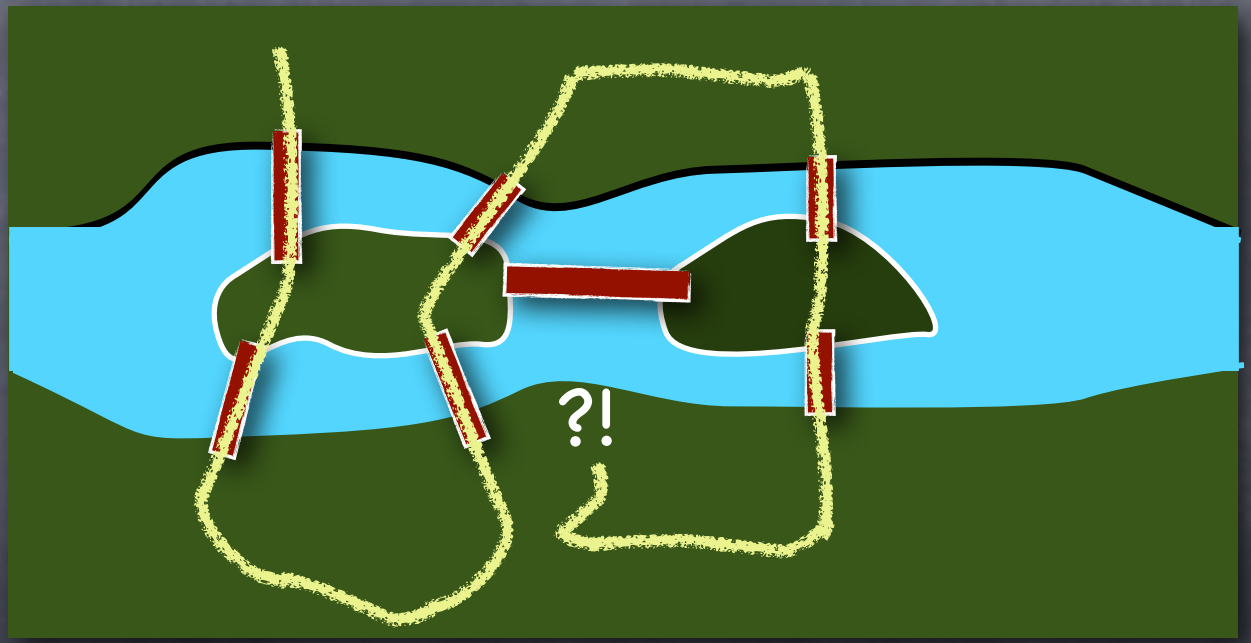
Bridges of Königsberg

- Cross each bridge exactly once



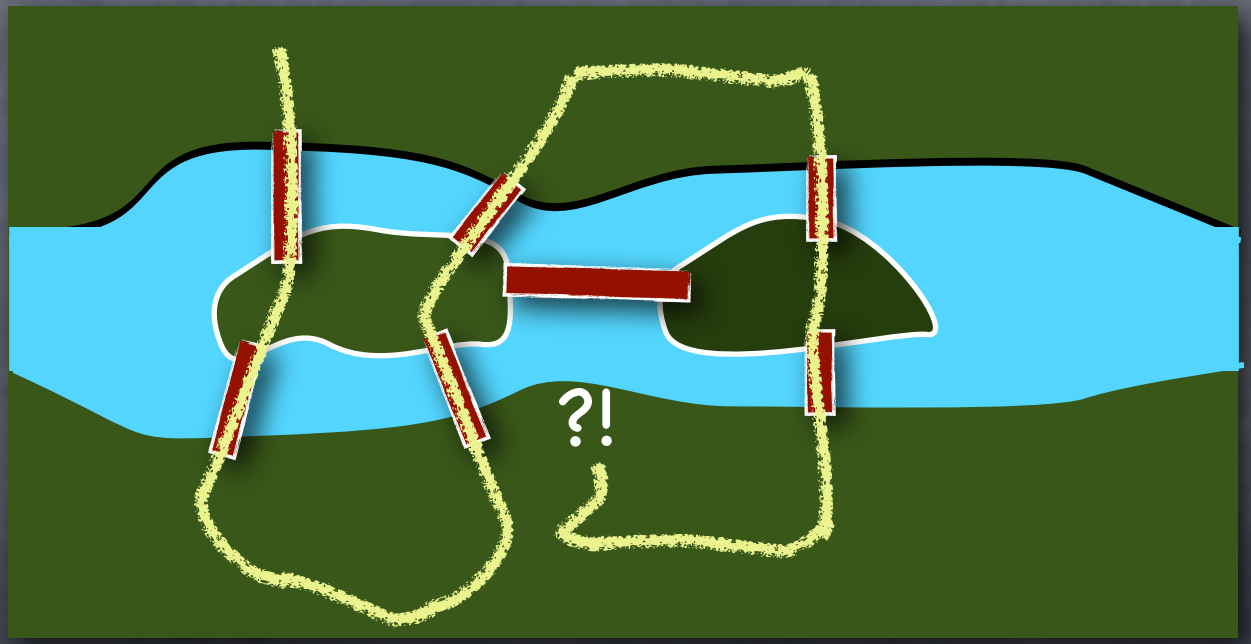
Bridges of Königsberg

- Cross each bridge exactly once



Bridges of Königsberg

- Cross each bridge exactly once



- Impossible! But how do we know for sure?

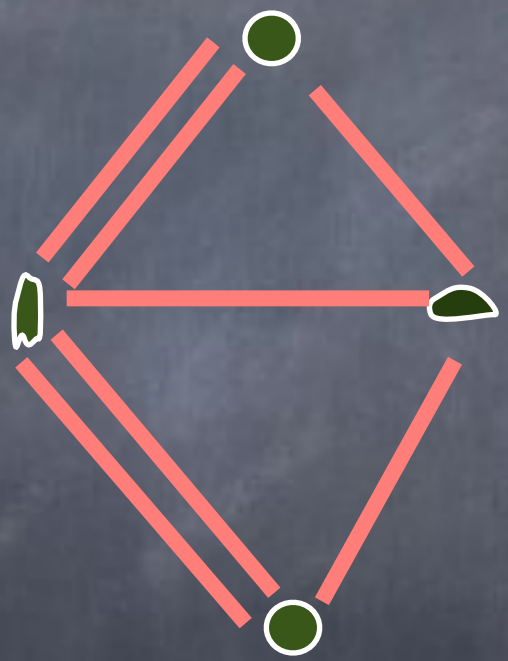
Bridges of Königsberg

- Cross each bridge exactly once

- Impossible! But how do we know for sure?

Bridges of Königsberg

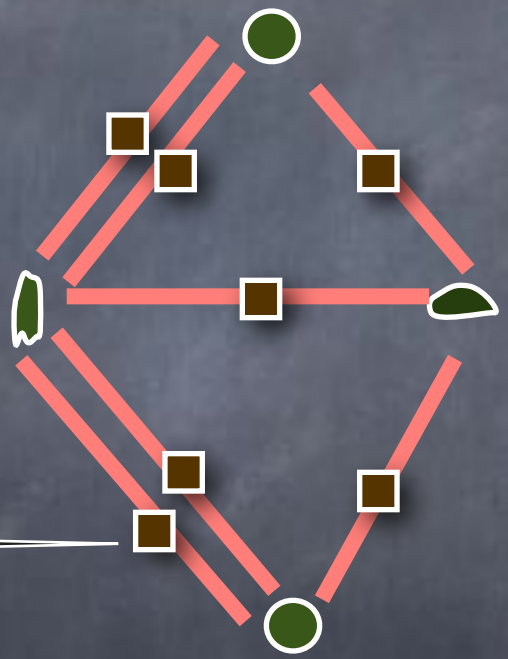
- Cross each bridge exactly once



- Impossible! But how do we know for sure?

Bridges of Königsberg

- Cross each bridge exactly once

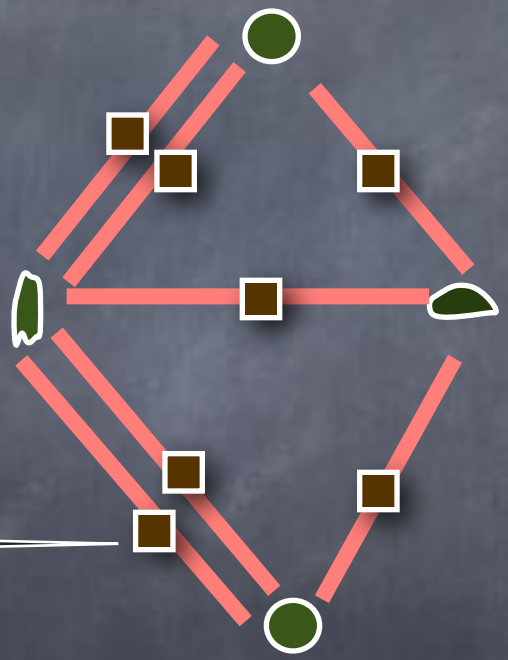


Add a node for each bridge too, if we want it to be a simple graph

- Impossible! But how do we know for sure?

Bridges of Königsberg

- Cross each bridge exactly once

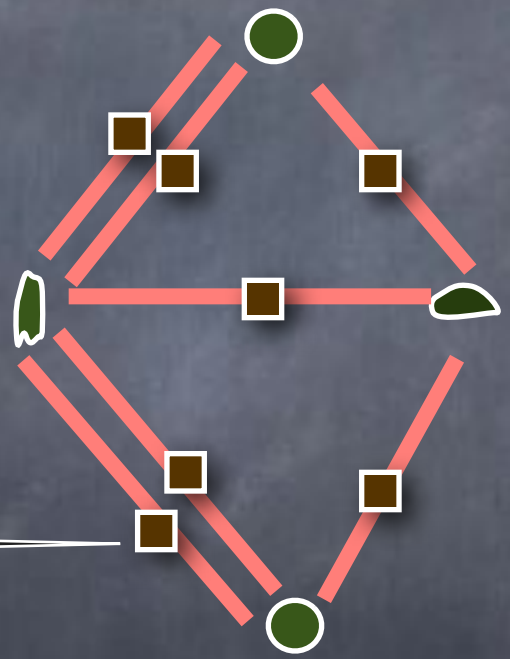


Add a node for each bridge too, if we want it to be a simple graph

- Impossible! But how do we know for sure?
- If there is a walk that takes each edge exactly once, then only the end nodes of the walk can have odd degree (why?)

Bridges of Königsberg

- Cross each bridge exactly once



Add a node for each bridge too, if we want it to be a simple graph

- Impossible! But how do we know for sure?
- If there is a walk that takes each edge exactly once, then only the end nodes of the walk can have odd degree (why?)

Eulerian Tour & Circuit

Eulerian Tour & Circuit

- Eulerian tour: a walk visiting every edge exactly once

Eulerian Tour & Circuit

- Eulerian tour: a walk visiting every edge exactly once
  - Eulerian tour exists → at most 2 odd degree nodes

Eulerian Tour & Circuit

- Eulerian tour: a walk visiting every edge exactly once
  - Eulerian tour exists → at most 2 odd degree nodes
    - Depending on whether the walk is open or closed

Eulerian Tour & Circuit

- Eulerian tour: a walk visiting every edge exactly once
  - Eulerian tour exists → at most 2 odd degree nodes
    - Depending on whether the walk is open or closed
- Eulerian circuit: a closed walk visiting every edge exactly once

Eulerian Tour & Circuit

- Eulerian tour: a walk visiting every edge exactly once
  - Eulerian tour exists → at most 2 odd degree nodes
    - Depending on whether the walk is open or closed
- Eulerian circuit: a closed walk visiting every edge exactly once
  - Eulerian circuit exists → no odd degree nodes

Question

Question

- Suppose G_1, G_2, G_3 are simple graphs with the following degree sequences: (2,2,2), (2,2,2,2,2,2), (0,0,2,2,2). Then which ones **must** have Eulerian circuits?

A. G_1
B. G_2
C. G_1 and G_2
D. G_1, G_2 and G_3
E. None of them

Question

- Suppose G_1, G_2, G_3 are simple graphs with the following degree sequences: (2,2,2), (2,2,2,2,2,2), (0,0,2,2,2). Then which ones **must** have Eulerian circuits?

A. G_1 — Only possibility is K_3
B. G_2
C. G_1 and G_2
D. G_1, G_2 and G_3
E. None of them

Question

- Suppose G_1, G_2, G_3 are simple graphs with the following degree sequences: (2,2,2), (2,2,2,2,2,2), (0,0,2,2,2). Then which ones **must** have Eulerian circuits?

A. G_1
B. G_2
C. G_1 and G_2
D. G_1, G_2 and G_3
E. None of them

Only possibility is K_3

Two possibilities: C_6 or the disjoint union of two K_3's.

Question

- Suppose G_1, G_2, G_3 are simple graphs with the following degree sequences: (2,2,2), (2,2,2,2,2,2), (0,0,2,2,2). Then which ones **must** have Eulerian circuits?

A. G_1
B. G_2
C. G_1 and G_2
D. G_1, G_2 and G_3
E. None of them

Only possibility is K_3

Two possibilities: C_6 or the disjoint union of two K_3's.

G_3 not connected

Eulerian Tour & Circuit

- Eulerian tour: a walk visiting every edge exactly once
  - Eulerian tour exists → at most 2 odd degree nodes
- Eulerian circuit: a closed walk visiting every edge exactly once
  - Eulerian circuit exists → no odd degree nodes

Eulerian Tour & Circuit

- Eulerian tour: a walk visiting every edge exactly once
  - Eulerian tour exists → at most 2 odd degree nodes
- Eulerian circuit: a closed walk visiting every edge exactly once
  - Eulerian circuit exists → no odd degree nodes
- If no odd degree nodes and connected, then must have an Eulerian circuit!

Eulerian Tour & Circuit

- Eulerian tour: a walk visiting every edge exactly once
  - Eulerian tour exists → at most 2 odd degree nodes
- Eulerian circuit: a closed walk visiting every edge exactly once
  - Eulerian circuit exists → no odd degree nodes
- If no odd degree nodes and connected, then must have an Eulerian circuit!
  - Informal argument: find and remove one cycle at a time (take a walk until repeated node) and finally stitch them all together into one Eulerian circuit