CIS529: Bioinformatics

Denovo Genome Assembly: Algorithmic Basis

Presented by

Dr. Fayyaz-ul-Amir Afsar Minhas

http://faculty.pieas.edu.pk/fayyaz

Department of Computer & Information Sciences
Pakistan Institute of Engineering & Applied Sciences
PO Nilore, Islamabad 45650
Pakistan

Basics

- **Hamiltonian Path**
- **Eulerian Path**
- **De Bruijn graphs**

Leonhard Euler
1805-1865

Eulerian path problem

William Rowan Hamilton
1805-1865

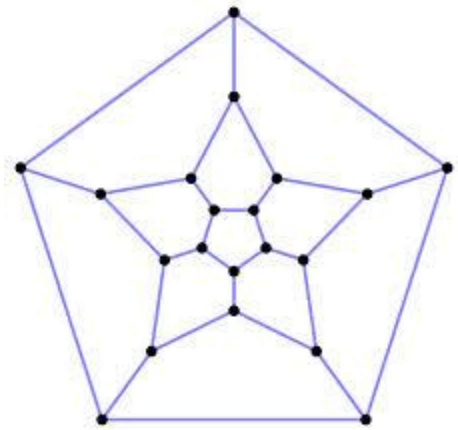
Hamiltonian path problem

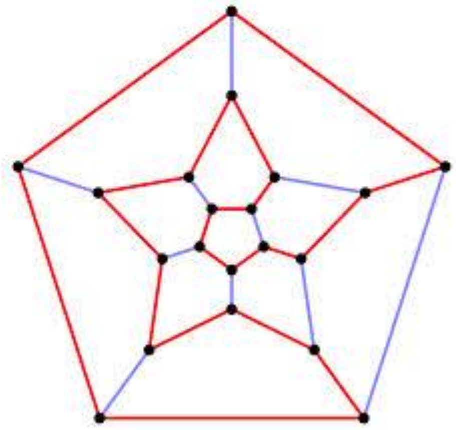
Nicolaas Govert de Bruijn
1918-2012

De Bruijn Graphs

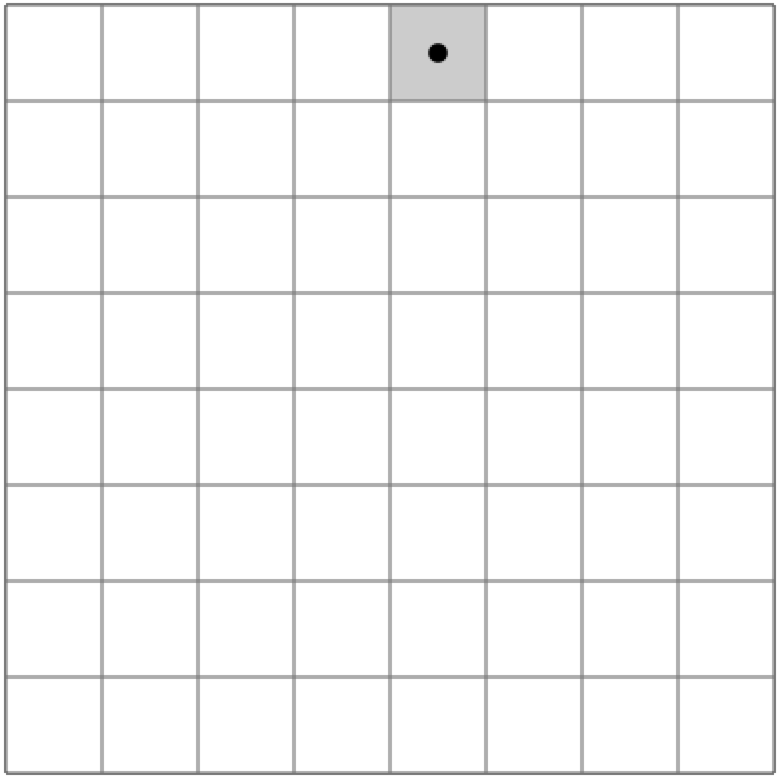
Hamiltonian Path Problem

- Icosian Game
  - In the figure on the right, find a cycle (a path that begins and ends at the same node) that visits each and all nodes exactly once
- Hamiltonian path
- Hamiltonian cycles
- Examples
  - Knight's tour problem
  - Traveling Salesman problem





Knight's tour

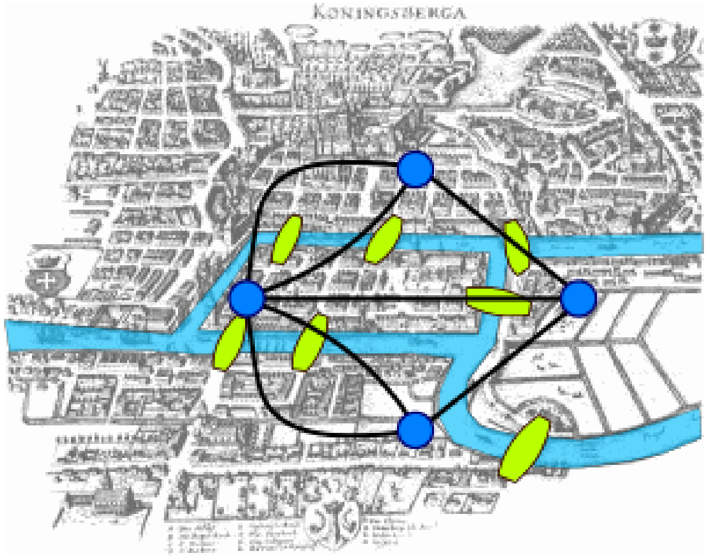


Hamiltonian path problem

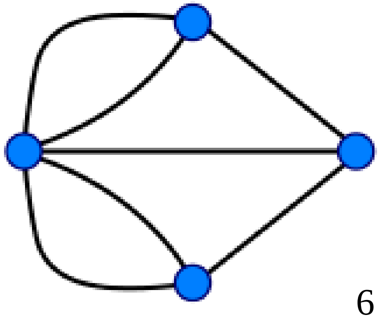
- Both determining the existence of a Hamiltonian cycle and finding it are NP-Complete
  - Time required to solve the problem using any currently known algorithm increases very quickly as the size of the problem grows.

Eulerian Path Problem

- **7 Bridges of Koningsberg**
  - **Find a path that goes over all bridges exactly once**



- **Eulerian path / walk / trail /cycle**
  - **Visits each edge exactly once**

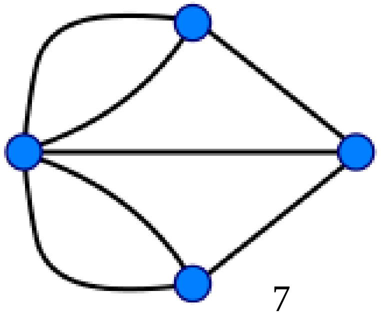


Eulerian Graphs

- A graph is said to be Eulerian if there is a Eulerian path in it
- Properties
  - An undirected graph has an Eulerian cycle if and only if all nodes of non-zero degree form a single connected component and have an even degree
  - An undirected graph has an Eulerian trail if and only if all nodes of non-zero degree form a single connected component and at most two nodes have odd degree

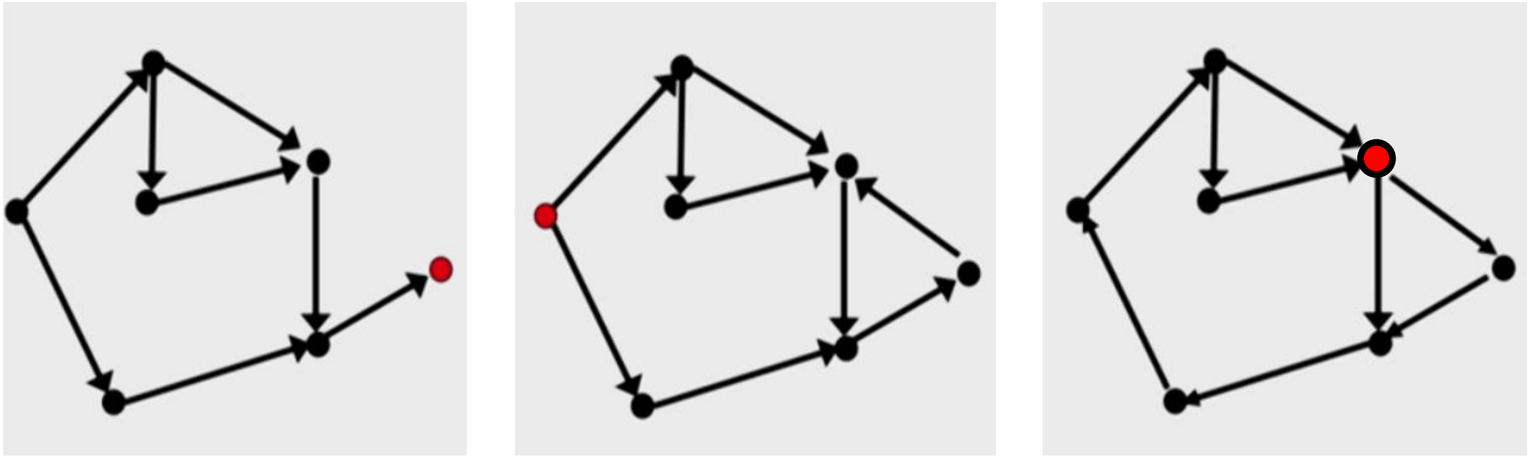
Does an Eulerian cycle exist in this graph? **NO**

Does an Eulerian path exist in this graph? **NO**



Eulerian Graphs

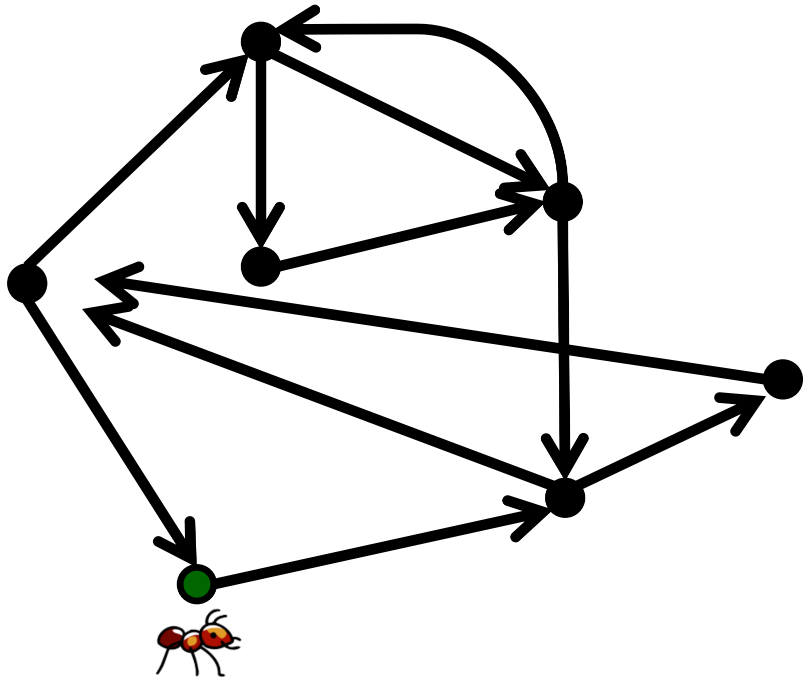
- Properties …
  - A directed graph has an Eulerian cycle iff every vertex has equal in-degree and out-degree, i.e., it's a balanced graph
- Is the following graph Eulerian?
  - No



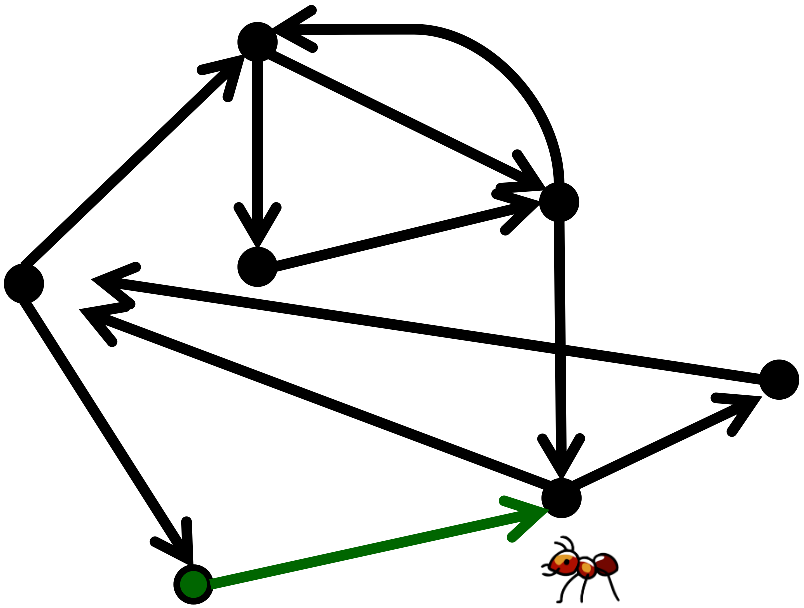
Finding an Eulerian Cycle

- An efficient algorithm exists for finding Eulerian cycle(s) in a balanced graph

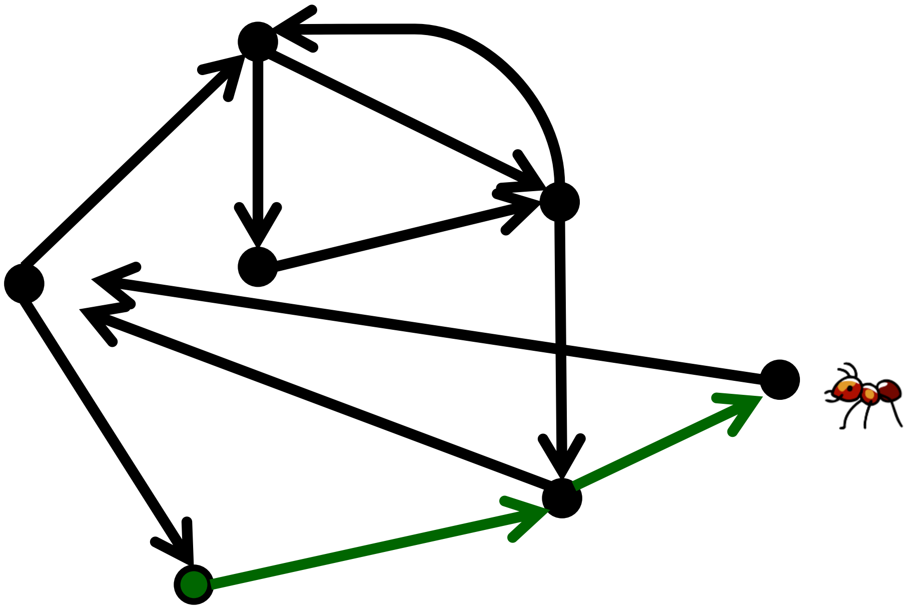
Finding Eulerian Cycles with Ants



Explore unexplored edges at random

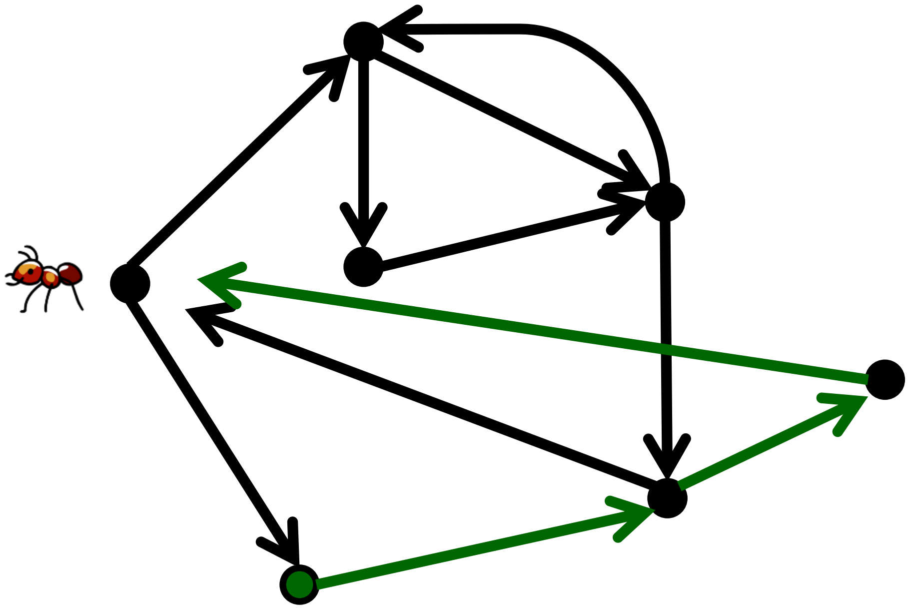


Exploring…



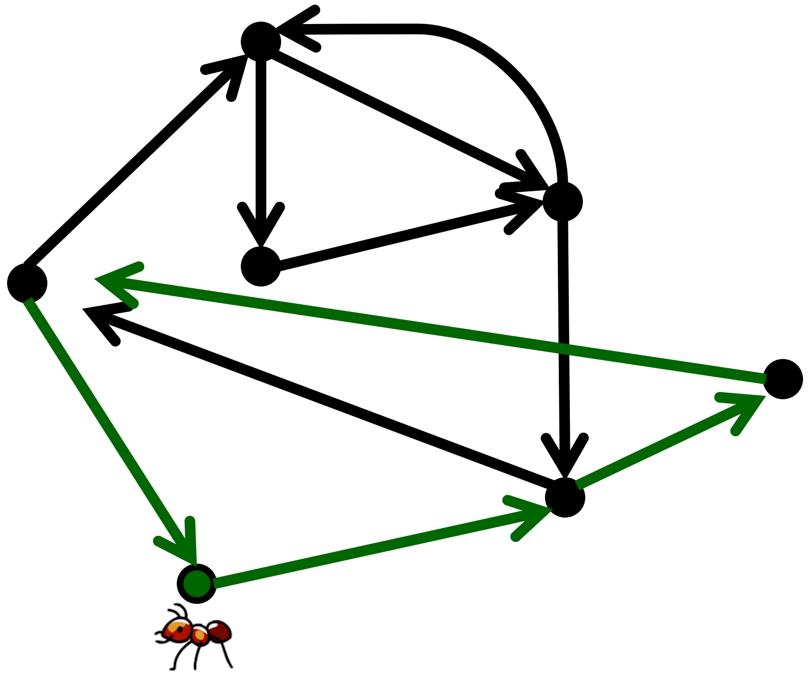
Walking …

- Can it get stuck? In what node?



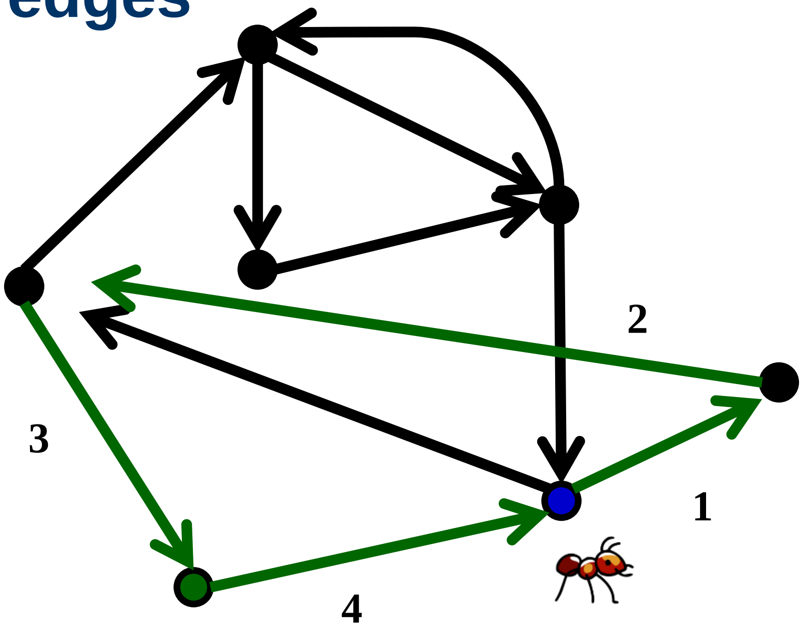
Back! But not solved!!

- **The ant will get stuck only in the starting node**



What to do now?

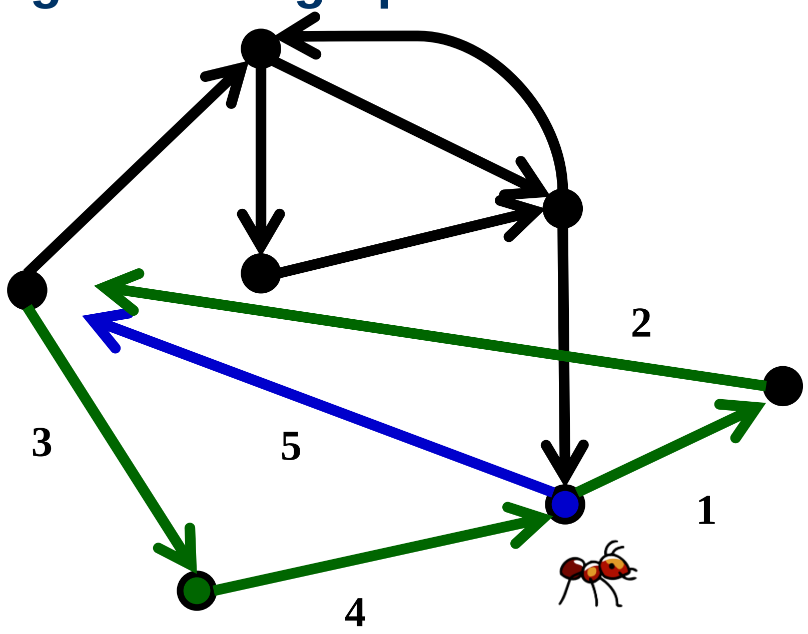
- Start at a node on the current cycle with still unexplored edges



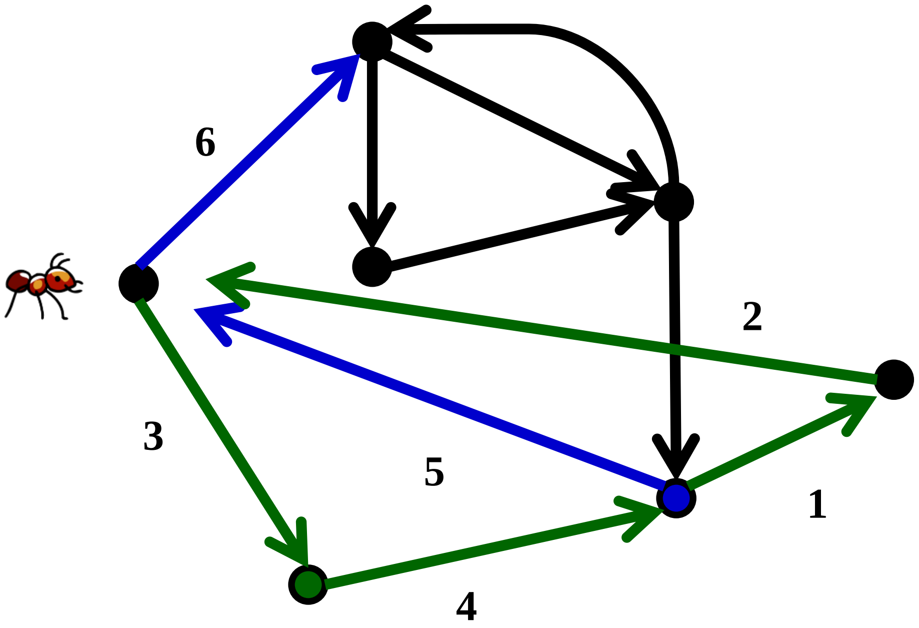
- Traverse all previously explored edges in the same order as before until you arrive back at the new start edge
  - Now the ant can continue because there is an edge!

Why repeat the cycle?

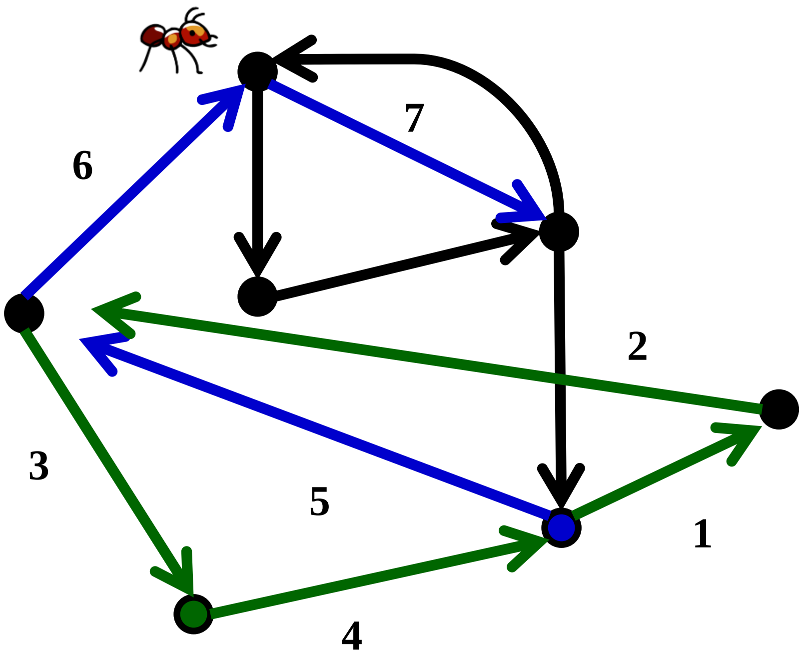
- After completing the cycle, start random exploration of untraversed edges in the graph



Walking …

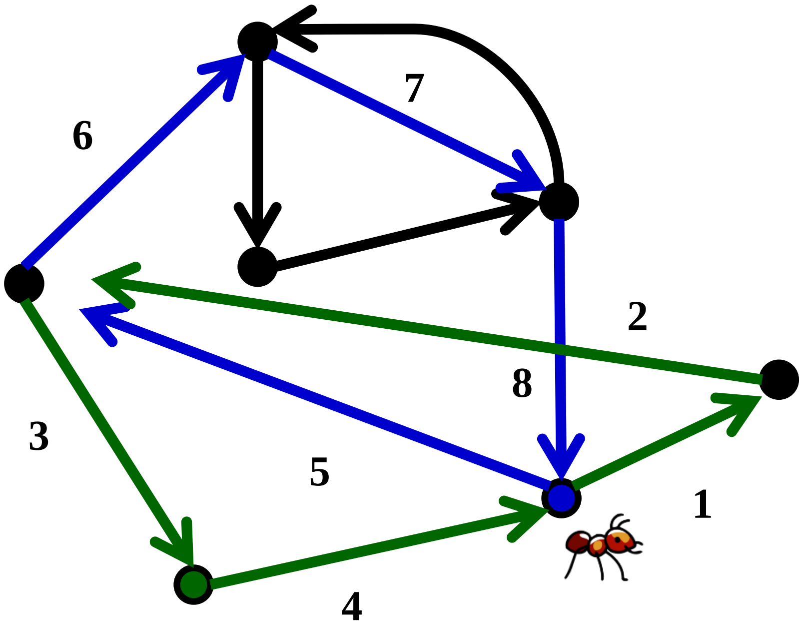


Walking …



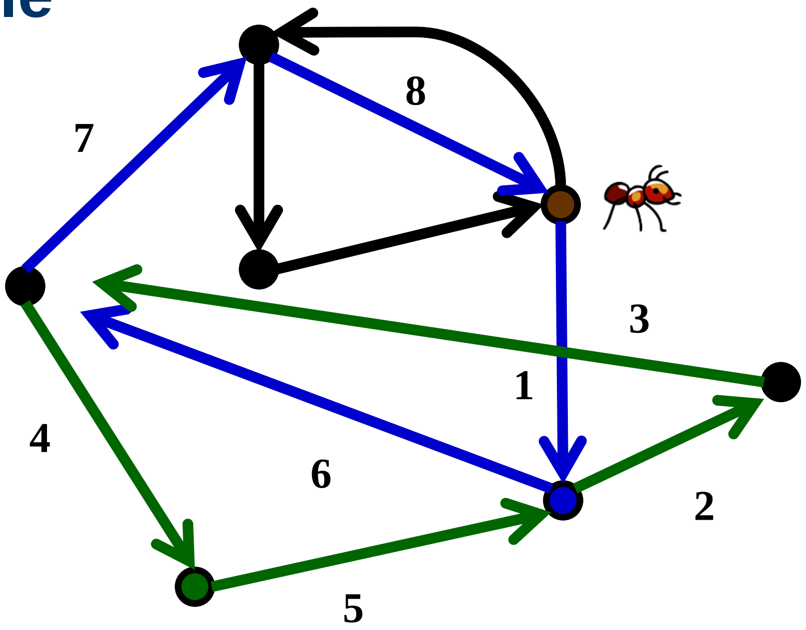
Walking ….

- Stuck Again

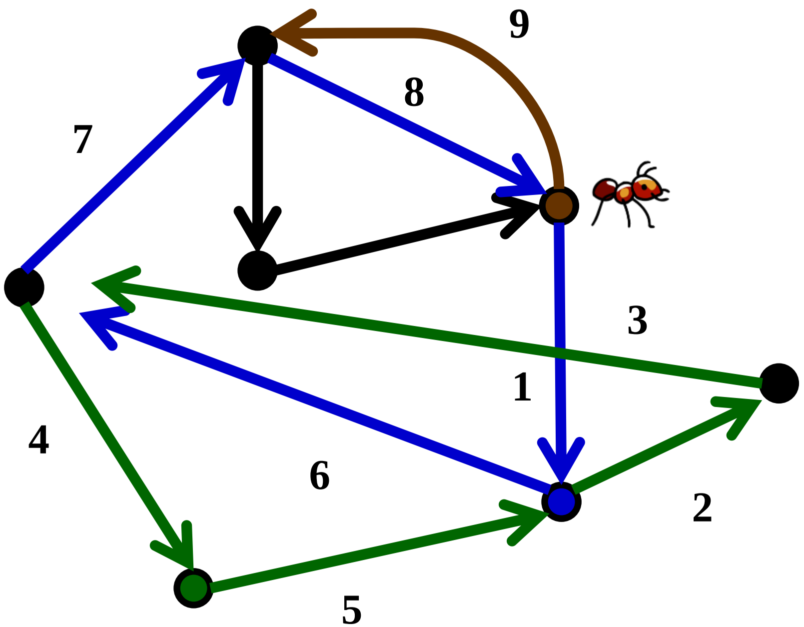


Stuck again!

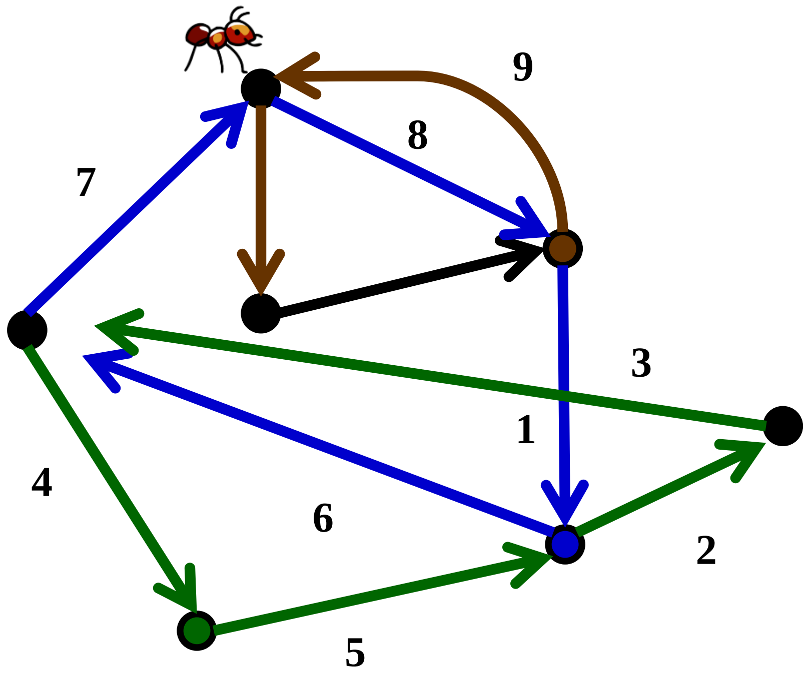
- Enlarge cycle by starting again and traversing the original cycle



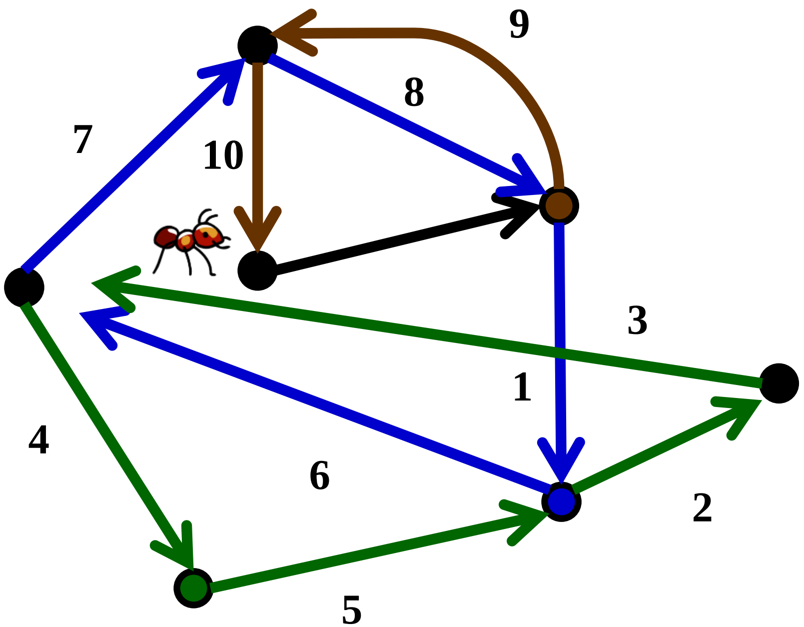
Completed the cycle! Walk again!



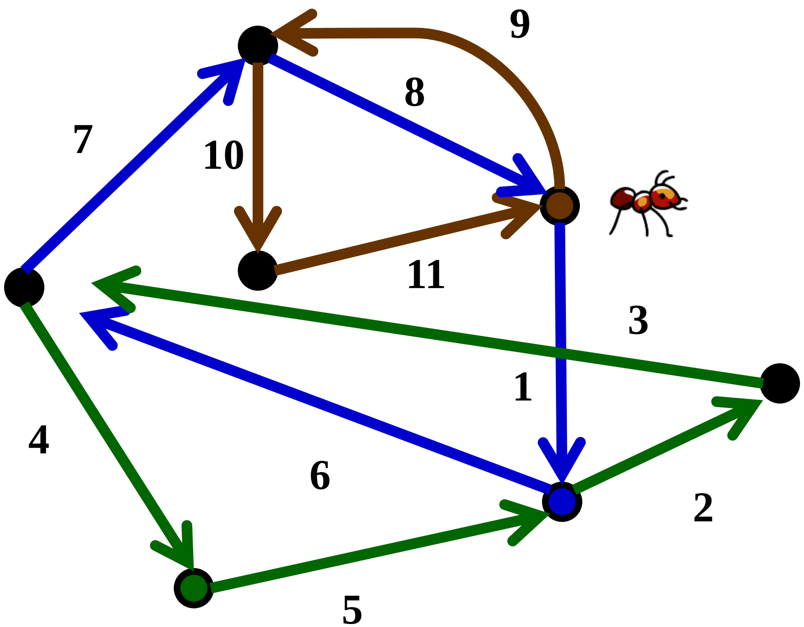
Walking!



Walking!

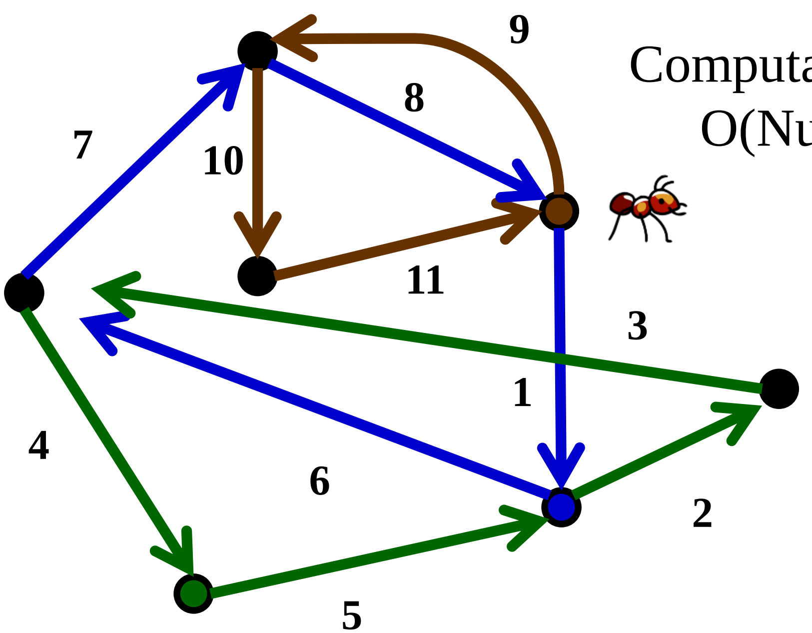


Done!



Algorithm

EulerianCycle(*BalancedGraph*)
  form a *Cycle* by randomly walking in *BalancedGraph* (avoiding already visited edges)
    while *Cycle* is not Eulerian
      select a node *newStart* in *Cycle* with still unexplored outgoing edges
      form a *Cycle'* by traversing *Cycle* from *newStart* and randomly walking afterwards
      Cycle ← *Cycle'*
  return *Cycle*



Computational complexity:
O(Number of edges)

k-Universal Circular String Problem

- The k-Universal Circular String Problem
  - Find a circular string containing each binary k-mer exactly once
  - Such strings are called De Bruijn Sequences
- Example
  - Given: k=3, 8 possible binary k-mers
  - Solution: 00011101
    - 000, 001, 011, 111, 110, 101, 010, 100
  - Given: k = 4, 16 possible binary k-mers
  - Solution: 0000110010111101
    - 0000, 0001, 0011, 0110, 1100, 1001, 0010, 0101, 1011, 0111, 1111, 1110, 1101, 1010, 0100, 1000

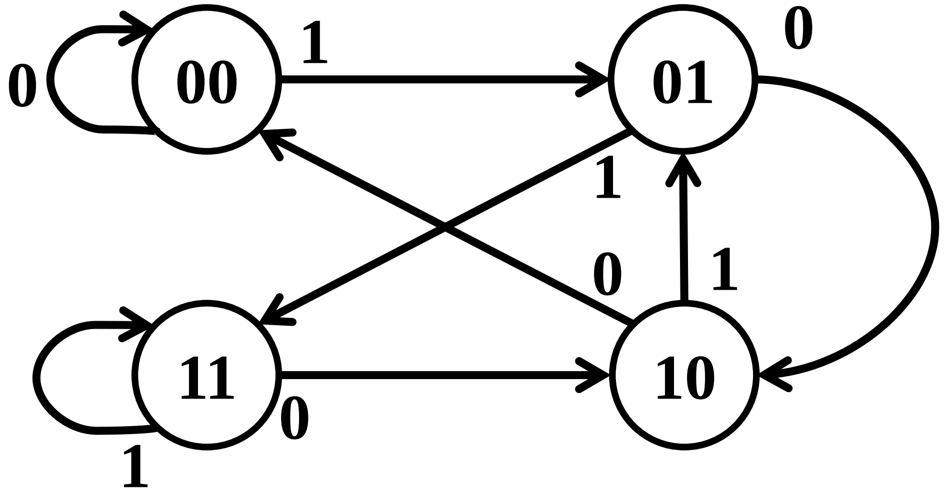
De Bruijn Graphs

- An (m,k) De Bruijn Graph is a graph defined over
  - An alphabet set S = $\{s_1, s_1, \ldots, s_m\}$
    - Example: S = {0,1} in case of binary strings (m = 2)
  - with the vertex set V = S^k = S x S x .. S
    - Example: V = {0,1} x {0, 1} = {(0,0), (0,1), (1,0), (1,1)} = {00, 01, 10, 11} (for k= 2)
  - and directional edges *a*→*b* between all nodes (a,b) such that *b* can be expressed in terms of *a* by shifting all its (*a*'s) symbols to the left and adding a new symbol from S at the end of this vertex
  - Example:
    - 00 → 01 is okay because if we shift 00 left and add 1 we get 01
    - 00 → 11 is not valid because 11 cannot be obtained by single shift

Exercise

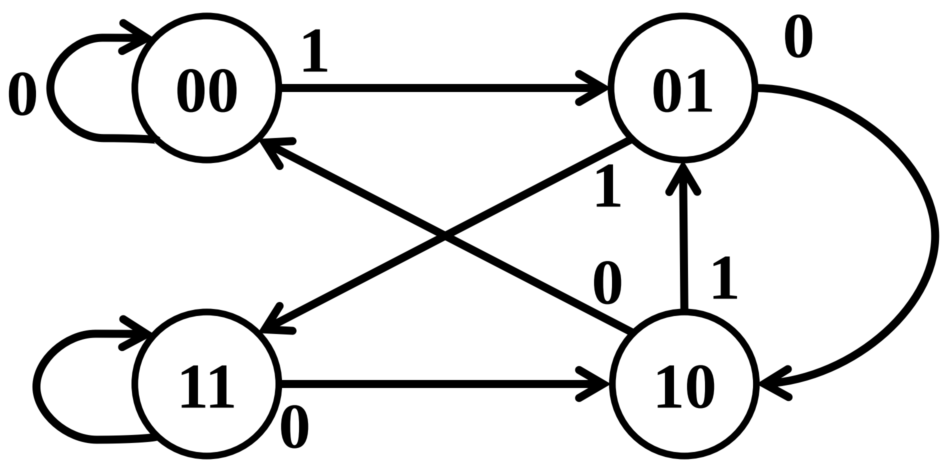
- Construct a (2,2) De Bruijn Graph with S = {0,1}

- Vertex Set: V = {00, 01, 10, 11}
- Edges:



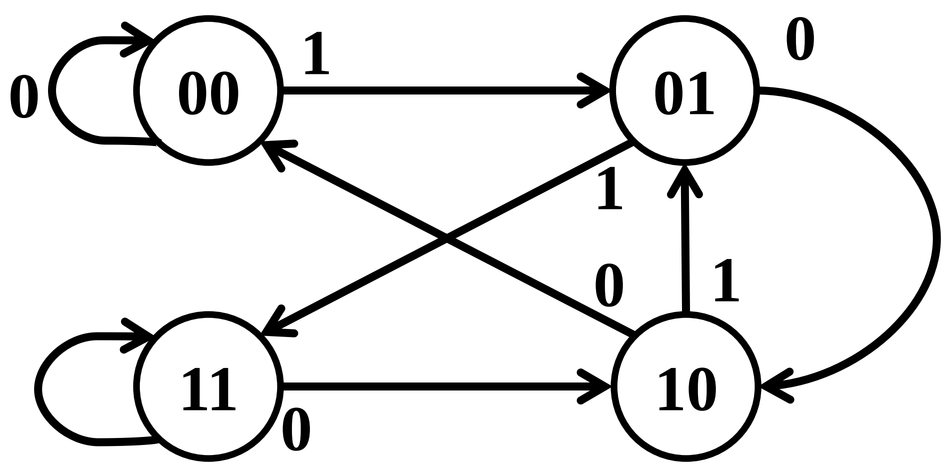
(2,2) De Bruijn Graphs

- Things to note
  - It's a balanced graph
    - Each vertex has m incoming and m outgoing edges



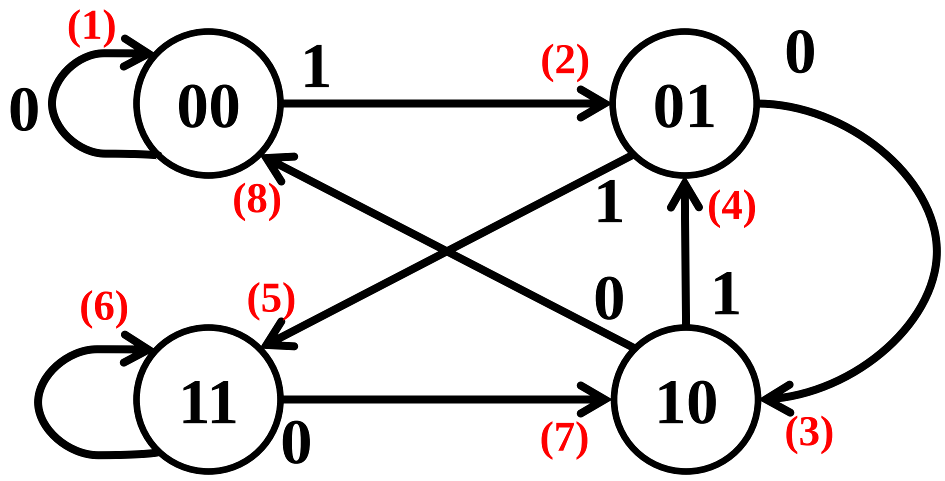
Hamiltonion cycle on (2,2) De Bruijn Graph

- Let's find the hamiltonian cycle in this graph
  - 00 → 01 → 11 → 10 → 00
  - All De Bruijn Graphs as Hamiltonian Graphs
  - Furthermore, the this cycle can generate the De Bruijn Sequence 0011 which the solution to the 2-Universal Circular String Problem



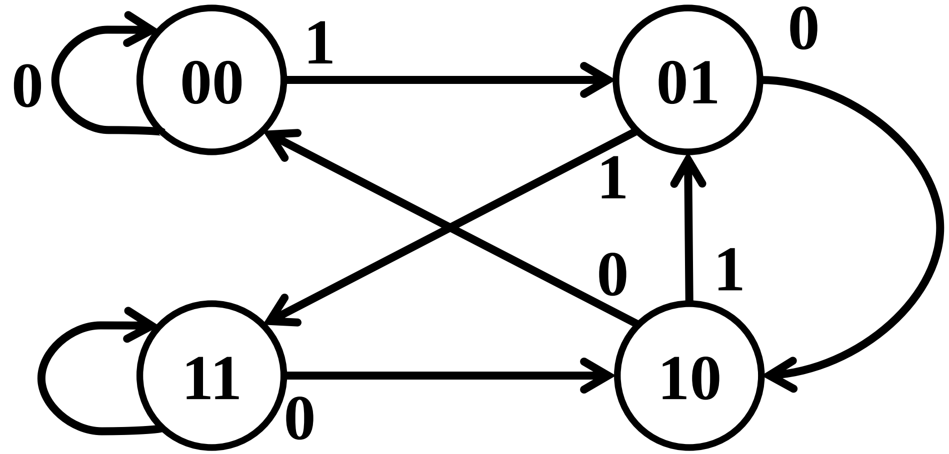
Eulerian cycle on (2,2) De Bruijn Graph

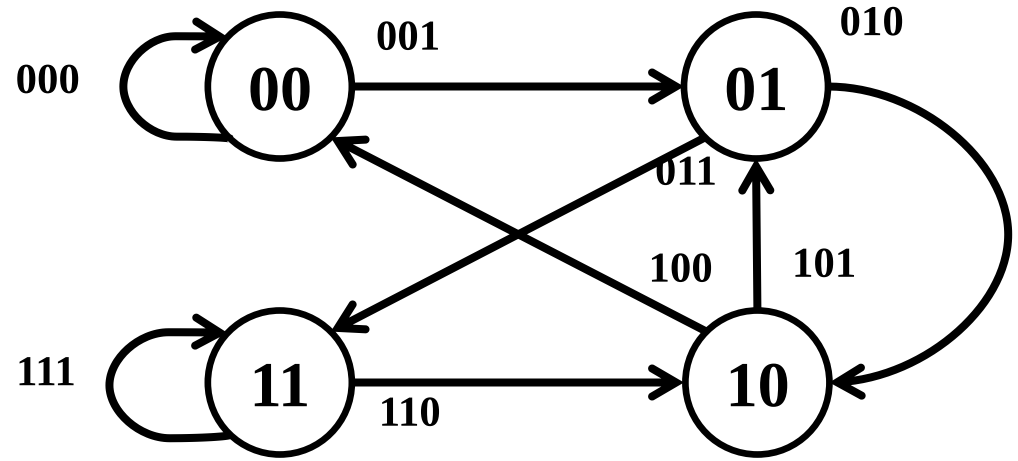
- Find a Eulerian Cycle on this graph
- All De Bruijn Graphs are Eulerian
- The solution 01011100 is the solution to the 3-Universal Circular String Problem
  - In general:
    - E(G(m,k)) = H(G(m,k+1)) = (k+1) binary De Bruijn Sequence



Alternative Labeling Style

- Edges can be labeled by the post-appending the source node with the original edge label

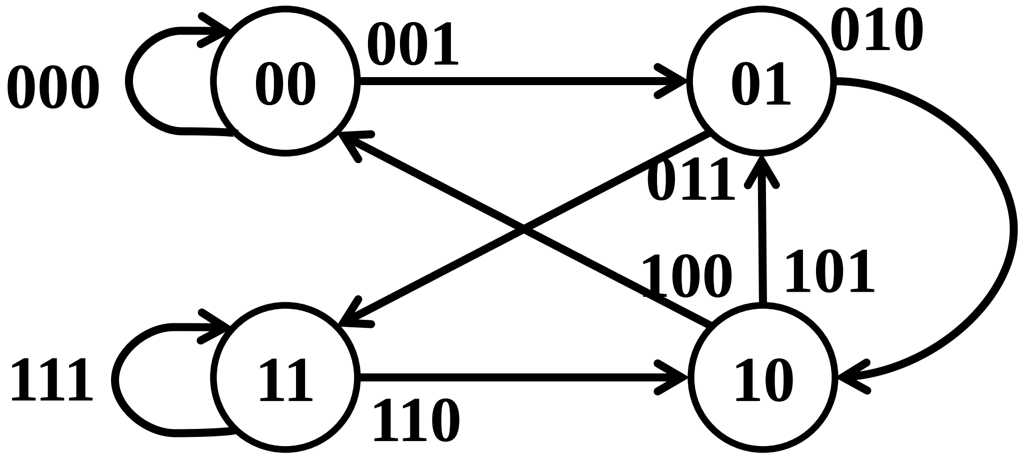




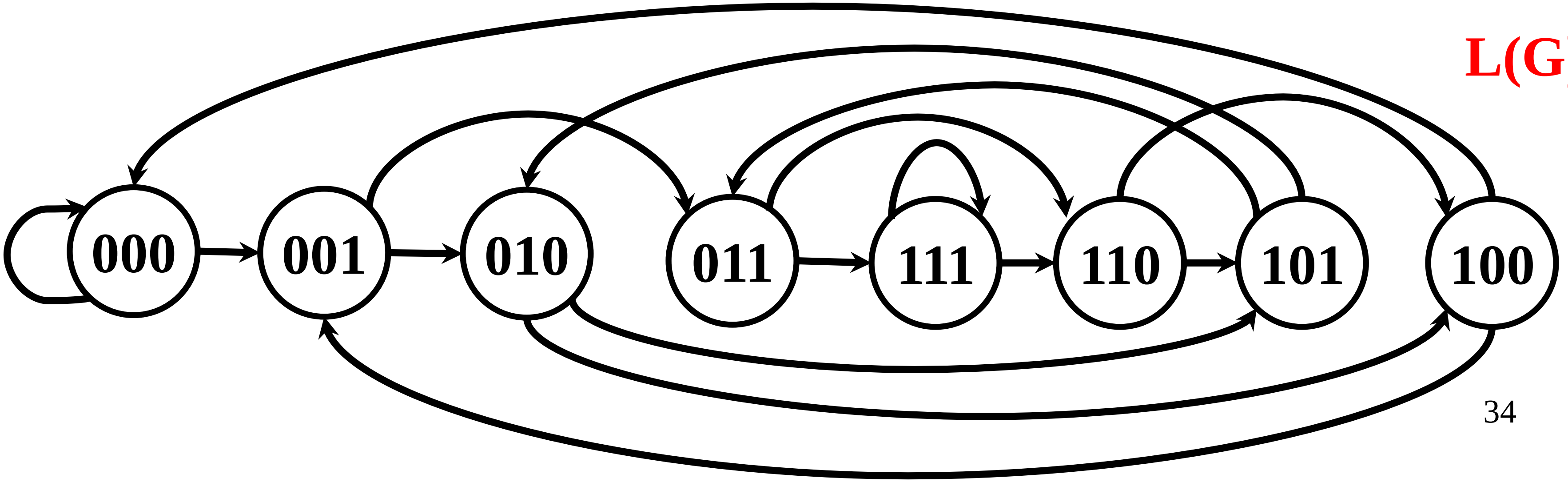
Line Graph Construction

- Given a graph G, its line graph L(G) is a graph such that
  - each vertex of L(G) represents an edge of G; and
  - two vertices of L(G) are adjacent if and only if their corresponding edges share a common endpoint ("are incident") in G.

Line Graph of (2,2) De Bruijn Graph



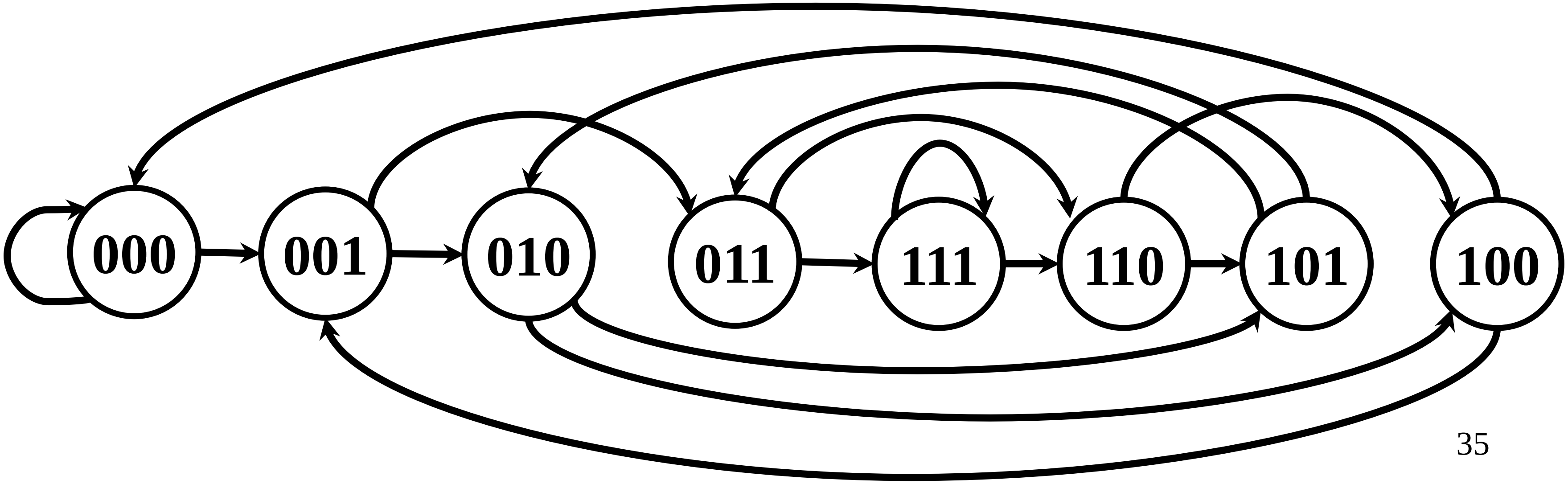
G

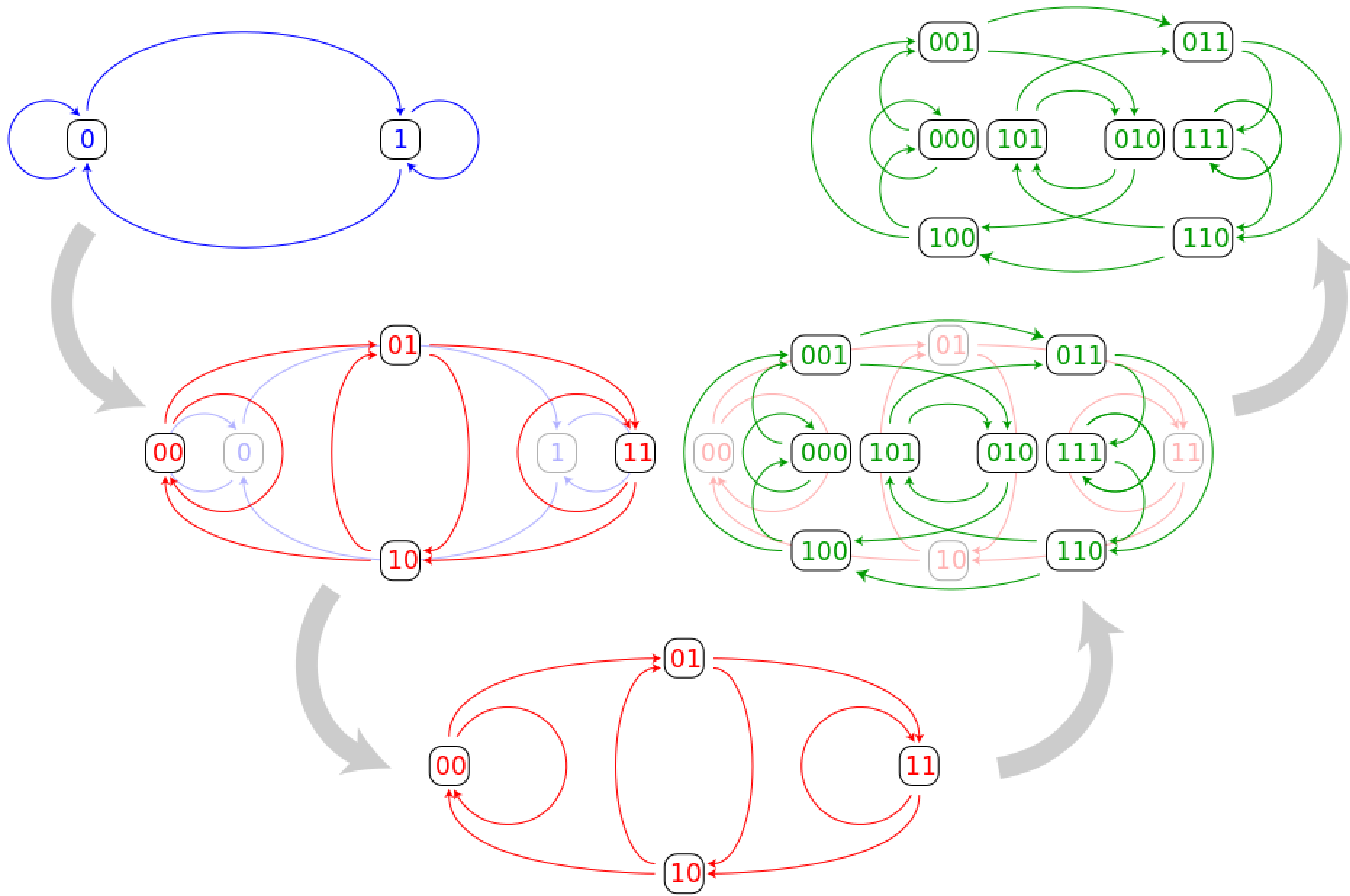


L(G)

Line Graphs of De Bruijn Graphs

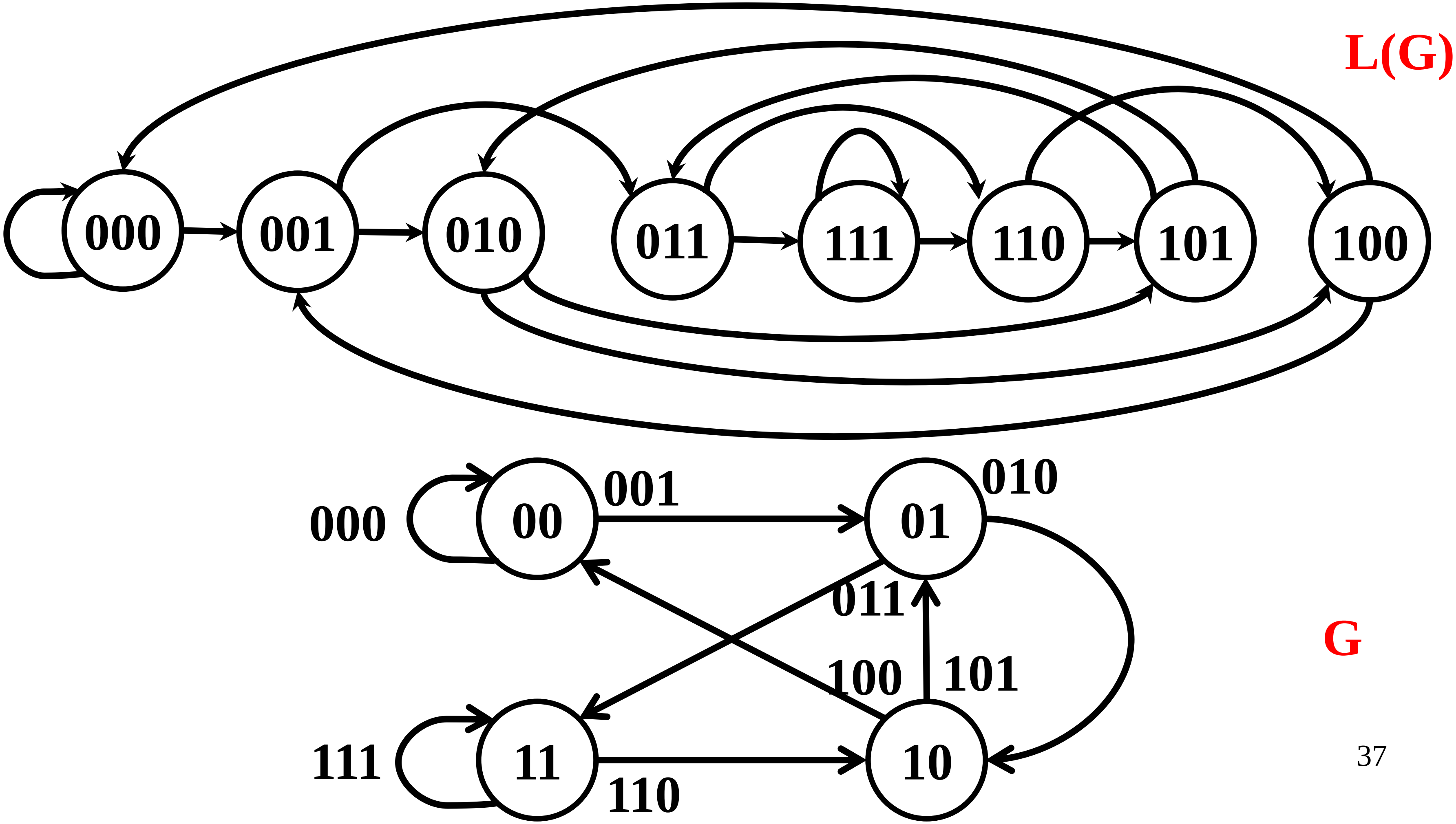
- Note that L(G(2,2)) is G(2,3)
- In general, for De Bruijn Graphs, L(G(m,k)) = G(m,k+1)
  - Since: E(G(m,k)) = H(G(m,k+1)) = (k+1) binary De Bruijn Sequence
  - Thus: E(G(m,k)) = H(L(G(m,k))) = (k+1) binary De Bruijn Sequence
- Thus, to find the solution to the k-Universal binary string problem, all we need is E(G(m,k-1))




0
1
001
011
000
101
010
111
100
110
01
00
0
1
11
10
001
01
011
00
000
101
010
111
11
100
10
110
01
00
11
10

Inverse Line Graphs

- Can we apply the operation in reverse?



Number of possible De Bruijn Sequences

- For (k,m), this number is
  - $\frac{(m!)^{m^{k-1}}}{m^k}$
  - For binary (m = 2)
    - $\frac{(2)^{2^{k-1}}}{2^k} = 2^{2^{k-1}-k}$
    - For k = 2
      - 1
    - For k = 3
      - 2
    - For k = 4
      - 16
    - For k = 5
      - 2048