2IT80 Discrete Structures

2023-24 Q2

Lecture 10: Directed graphs

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Recap

Graph

A graph G is an ordered pair (V, E), where

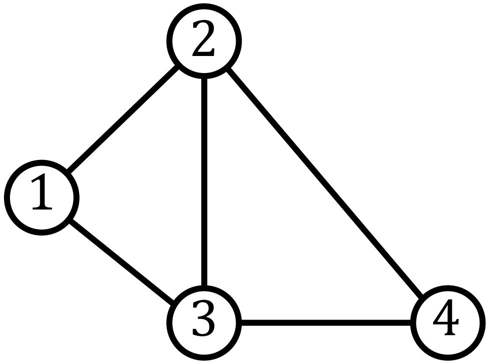
> V is a set of elements, called vertices.
> E a set of 2-element subsets of V, called edges

The degree of a vertex is equal to the number edges it is part of.

Vertices $v, v' \in V$ are adjacent when $\{v, v'\} \in E$. We say v' is a neighbor of v (and v a neighbor of v').

Example:
$V = \{1, 2, 3, 4\}$
$E = \{\{1,2\}, \{1,3\}, \{2,3\}, \{2,4\}, \{3,4\}\}$
Degree of vertex 2 is three.

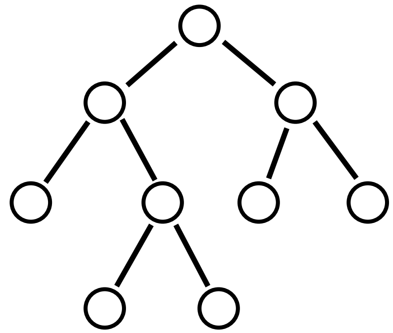
Vertex 2 and vertex 3 are adjacent.



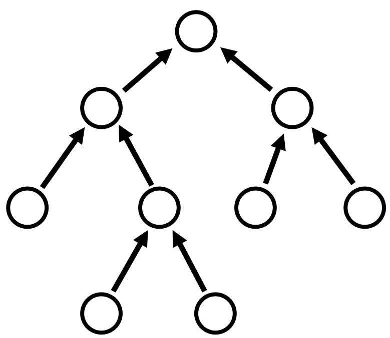
Directed graphs

Rooted trees

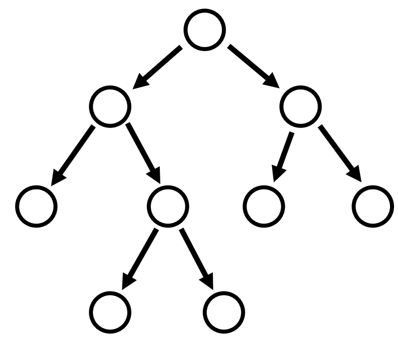
- Edge of a rooted tree have an implicit direction



Towards root

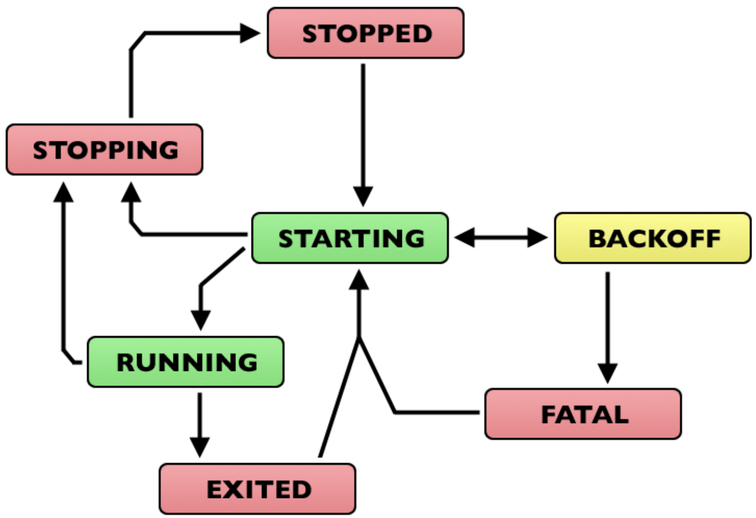


Away from root



What about other graphs?
Any examples of graphs where edges should have a direction?

Graphs



Directed graph

A directed graph G is an ordered pair (V, E), where V is some set of elements and E is a set of ordered pairs of V.

A directed edge $e = (x, y)$, called an *edge from x to y*, has head y and tail x.

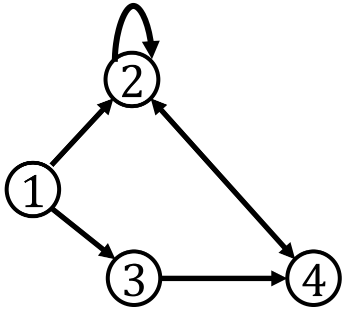
The indegree $\deg^+_G(v)$ of a vertex v is the number of edges having v as head. The outdegree $\deg^-_G(v)$ is the number of edges having v as tail.

Example:

$$V = \{1, 2, 3, 4\}$$
$$E = \{(1{,}2), (1{,}3), (2{,}2), (2{,}4), (3{,}4), (4{,}2)\}$$

Indegree of vertex 2 is three.

Outdegree of vertex 3 is one.

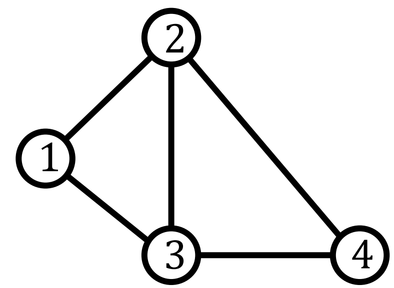


Graphs versus Directed graph

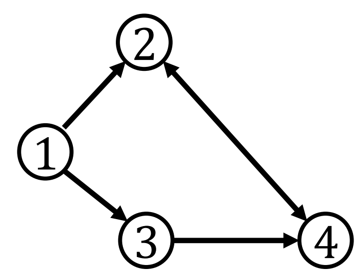
- In this course a Directed graph is NOT a Graph

- “Draw a graph with 7 edges and 5 vertices”
  - Graph, not directed graph!

Graph
$V = \{1, 2, 3, 4\}$
$E = \{\{1,2\}, \{1,3\}, \{2,3\}, \{2,4\}, \{3,4\}\}$



Directed graph
$V = \{1, 2, 3, 4\}$
$E = \{(1,2), (1,3), (2,4), (3,4), (4,2)\}$



Connectedness

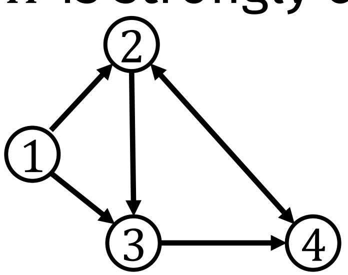
The symmetrization $sym(G)$ of a directed graph $G = (V, E)$ is defined by $sym(G) = (V, \bar{E})$, where $\bar{E} = \{\{x, y\}: (x, y) \in E \text{ or } (y, x) \in E\}$

A directed graph G is called weakly connected, when $sym(G)$ is connected.

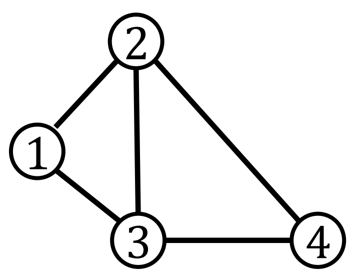
A directed graph $G = (V, E)$ is strongly connected when for every two vertices $v, v' \in V$ there is a directed path from v to v' and a directed path from v' to v.

Example: G is weakly connected

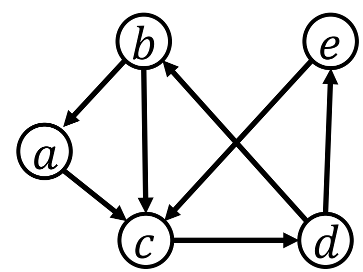
H is strongly connected



$G = (V, E)$



$sym(G) = (V, \bar{E})$

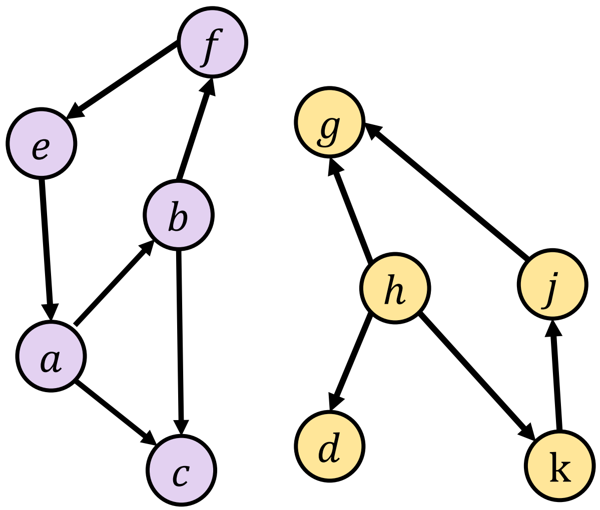


$H = (V, E)$

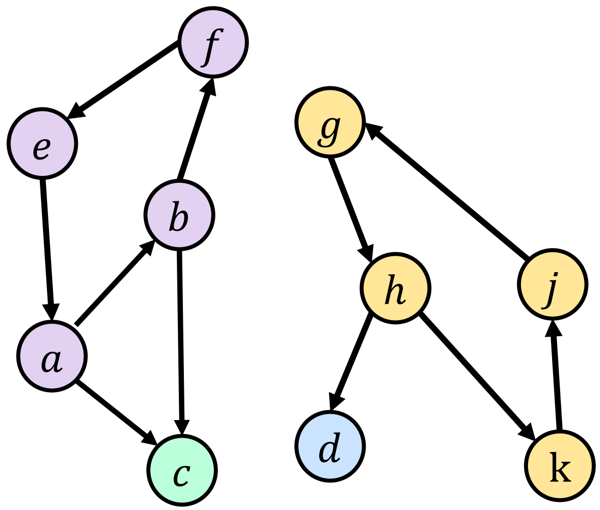
Connected components

Weakly connected components of a directed graph G are the equivalence classes defined by the relation $\sim$ on the set $V(G)$, where $x{\sim}y$ if and only if there exists a walk from x to y in the symmetrization of G.

Strongly connected components of a directed graph G are the equivalence classes defined by the relation $\sim$ on the set $V(G)$, where $x{\sim}y$ if and only if there exists a directed walk from x to y and from y to x in G.



Weakly connected components

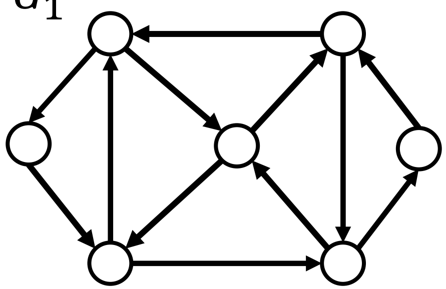


Strongly connected components

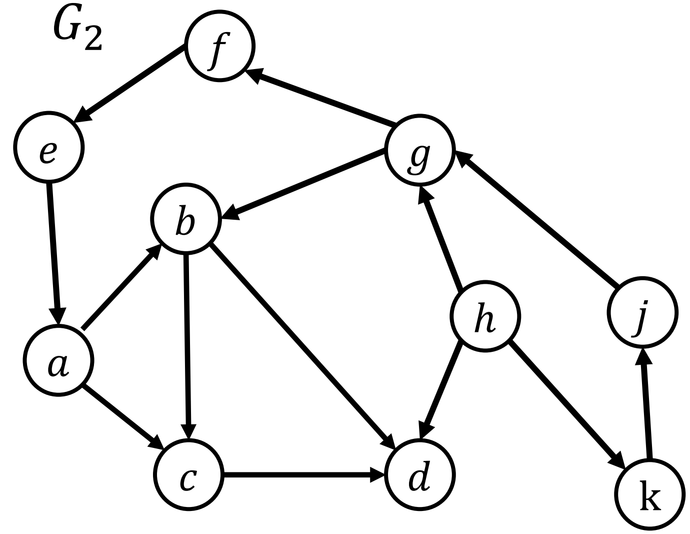
Quiz

For each of the following graph determine the number of strongly and weakly connected components:

G_1



G_2



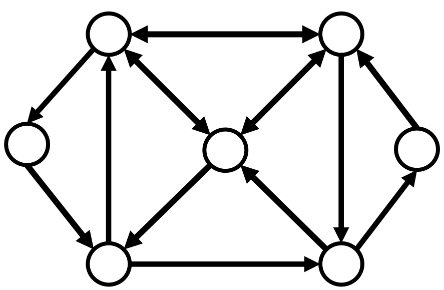
$$G_3 = (V_1, E_1)$$
$$V_1 = \{AMS, LDN, NRT, LAX\}$$
$$E_1 = \{(AMS, LDN), (NRT, LAX), (LAX, AMS), (NRT, AMS), (LDN, AMS)\}$$

Eulerian directed graph

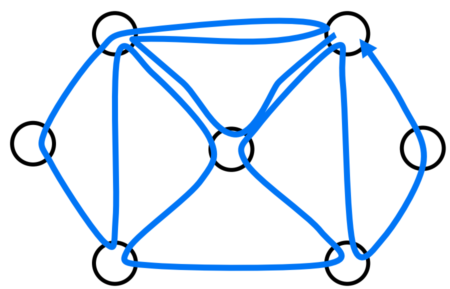
A closed directed walk containing all the vertices and edges, and each edge exactly once is a Eulerian directed tour.

A (directed) graph possessing a closed (directed) Eulerian tour is an Eulerian directed graph.

Example:

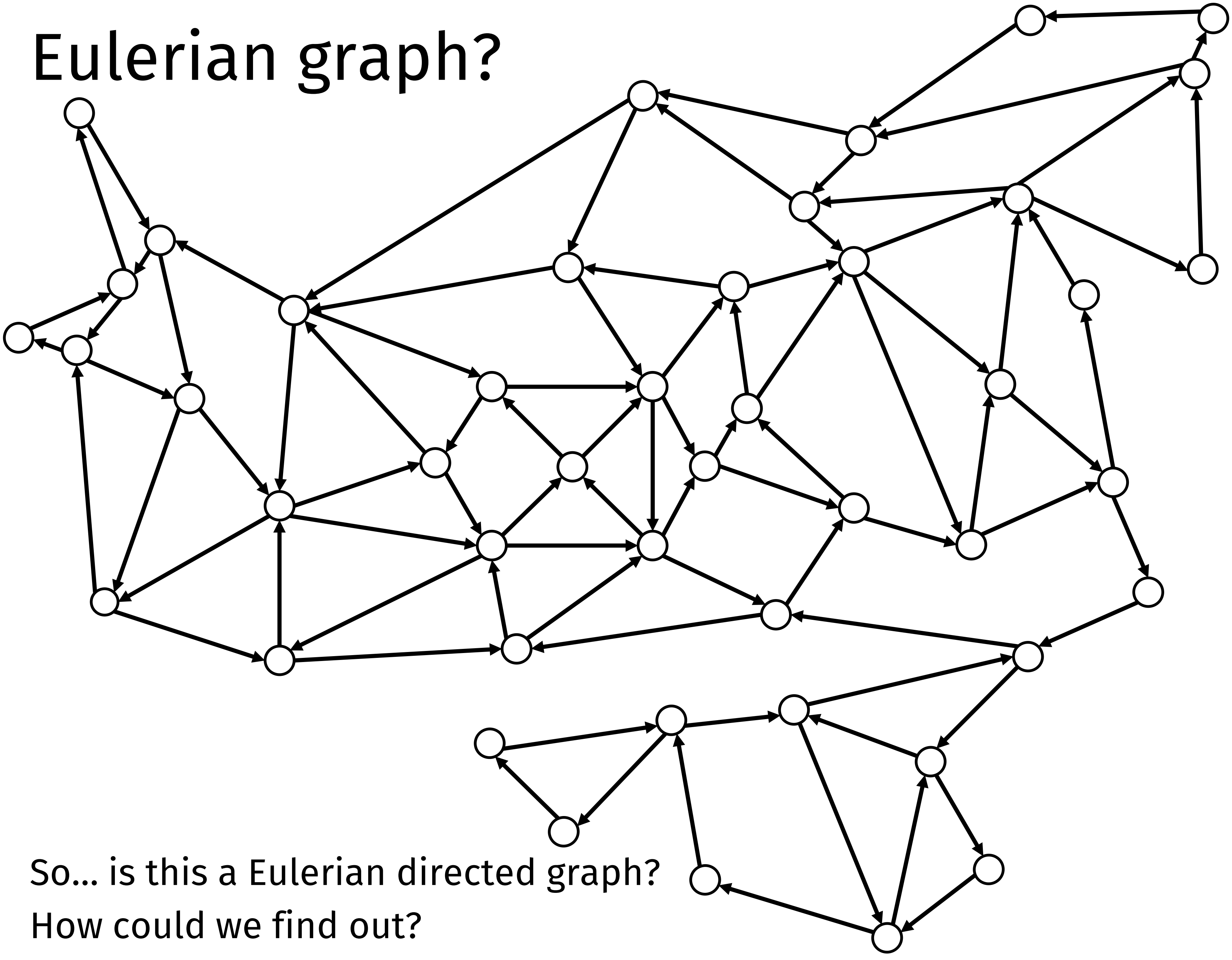


Eulerian graph



Closed Eulerian tour

Eulerian graph?



So... is this a Eulerian directed graph?

How could we find out?

Eulerian graph

Theorem: A directed graph is Eulerian if and only if its symmetrization is connected and $\deg^+ G(v) = \deg^- G(v)$ holds for each vertex $v \in V(G)$.

Proof?

Exercise!

Let's do an application for a change.

We have a wheel with binary digits on it.

Due to a slot we can see k digits at a time.

7	8	9
4	5	6
1	2	3
	0	

The keypad problem

You are staying at an Airbnb and the host has sent you the 4-digit code to the keypad. However, you realize you have no internet and no way to access the code.

Given that you are tired you want to be able to enter the house as quickly as possible.

How many combinations are there?

There is no “OK” button, as long as the last 4 digits you entered are the code you can enter.

At most how many numbers do you have to type?

7	8	9
4	5	6
1	2	3
	0	

The keypad problem

Let’s simplify and generalize.

Only two keys, 0 and 1

Code has length k

0	**1**

Keys we press form a sequence and we want to ensure it contains the code as a subsequence.

Example: code 011001

Sequence: 011010101001001001011010110010000111101

Problem: What is a minimal length of a sequence of 0’s and 1’s that contains every sequence of length k as a subsequence.

The keypad problem

Problem: What is a minimal length of a sequence of 0’s and 1’s that contains every sequence of length k as a subsequence.

Upper bound

Writing out all subsequences gives length $k2^k$

Can we do better?

The keypad problem

Problem: What is a minimal length of a sequence of 0's and 1's that contains every sequence of length k as a subsequence.

Upper bound

Writing out all subsequences gives length $k2^k$

Lower bound:

a sequence of length $2^k + k - 1$ contains exactly 2^k subsequences of length k

To make a sequence of this length, no subsequences can be repeated.

Problem: Find a sequence of maximal length in which no subsequence of length k is repeated.

Tuples

Let’s think about the problem.

How could we model this?

$$\boxed{001}10$$

Tuples

Let's think about the problem.

How could we model this?

$0\boxed{011}0$

Tuples

Let’s think about the problem.

How could we model this?

$00\boxed{110}$

Tuples

Let’s think about the problem.

How could we model this?

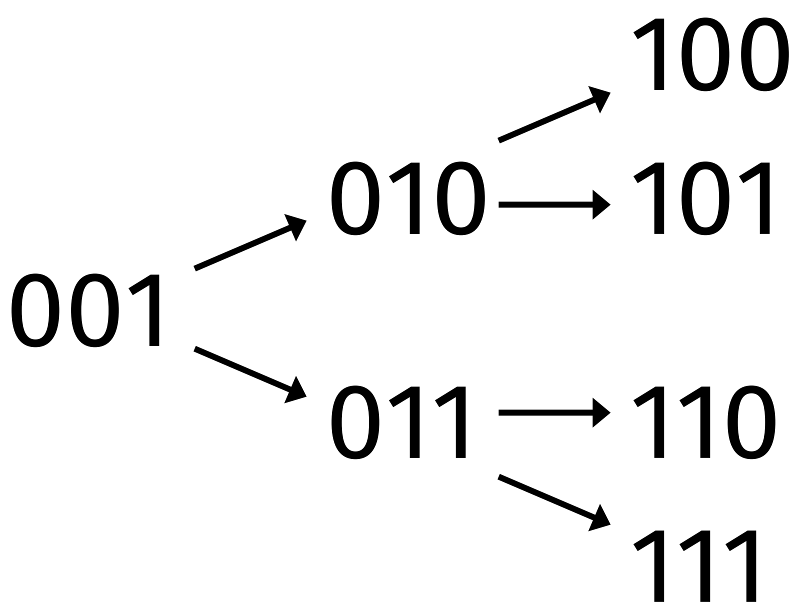
$\boxed{001}??$

$$
\begin{tikzcd}
 & 010 \\
001 \arrow[ur] \arrow[dr] & \\
 & 011
\end{tikzcd}
$$

Tuples

Let's think about the problem.

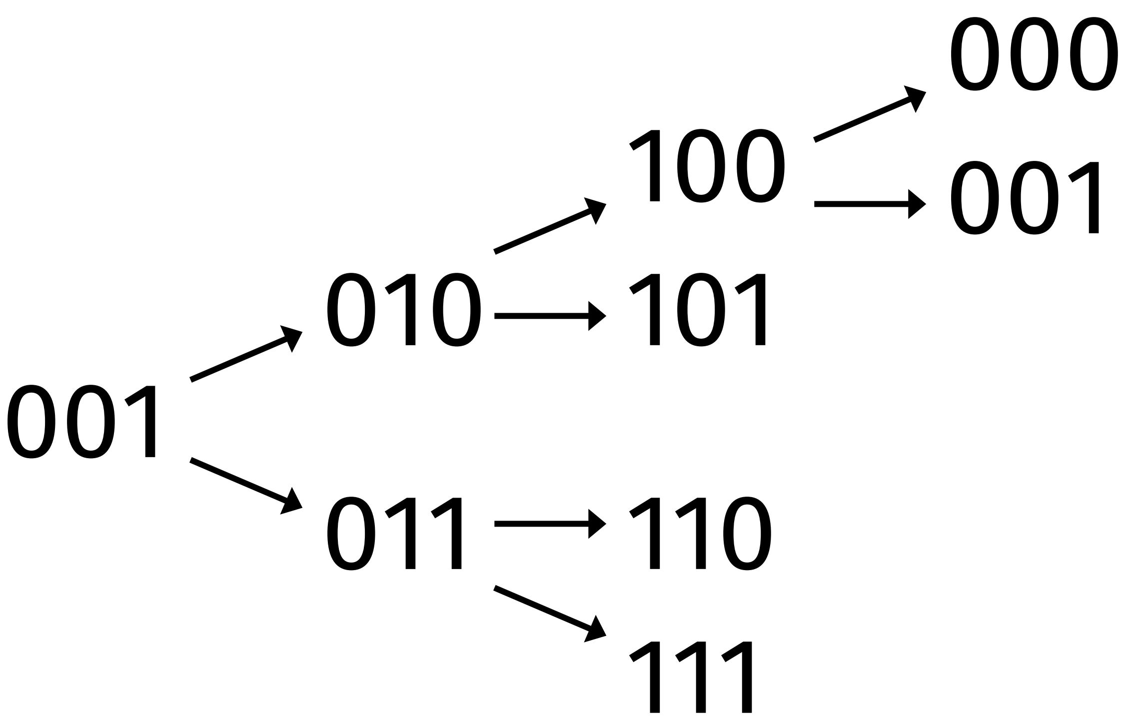
How could we model this?



Tuples

Let's think about the problem.

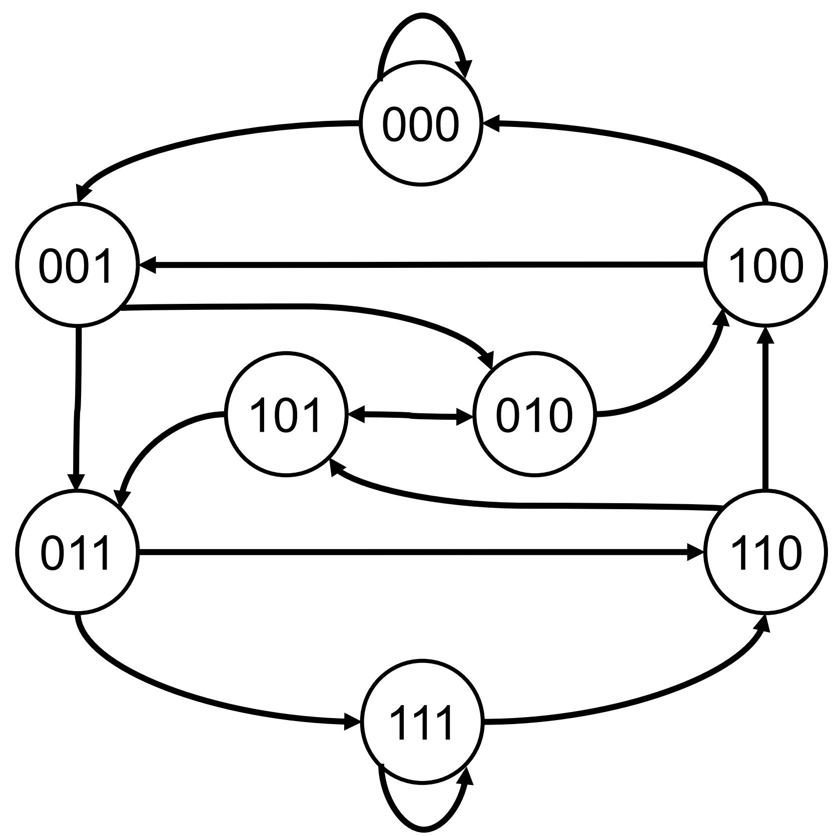
How could we model this?



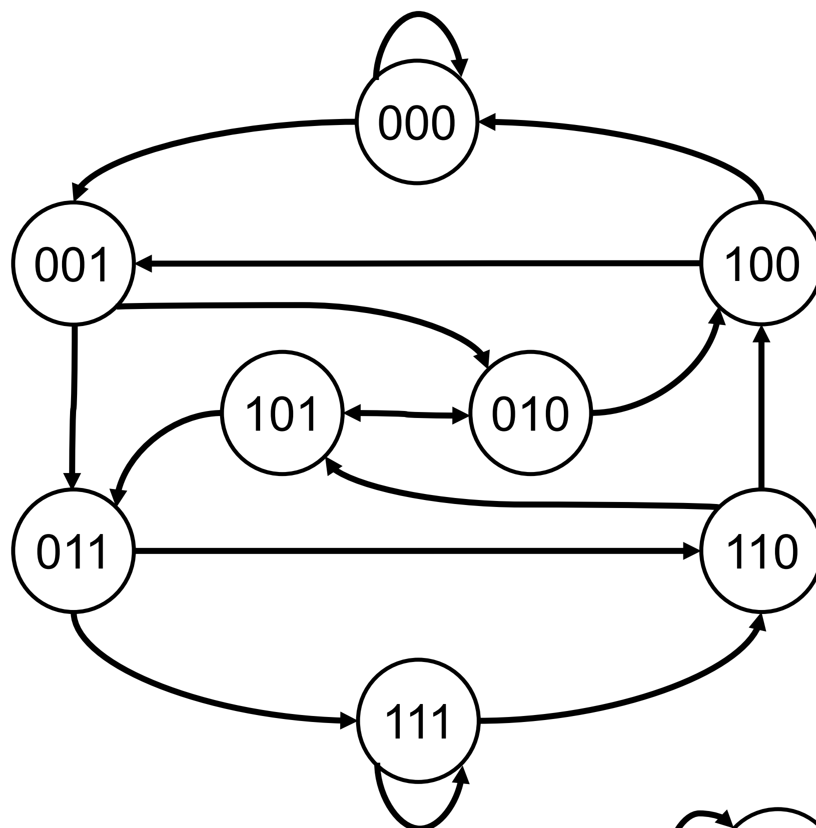
Digit graphs

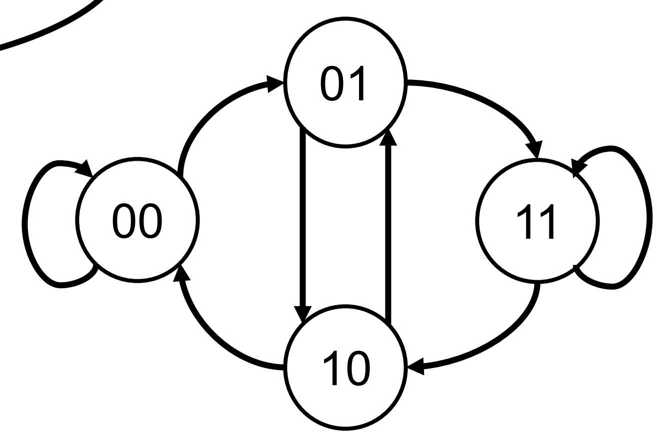
V: binary sequences of length k

E: the set with $((a_1, \ldots, a_k), (a_2, \ldots, a_{k+1}))$ for any $a_1, .., a_{k+1} \in \{0,1\}$



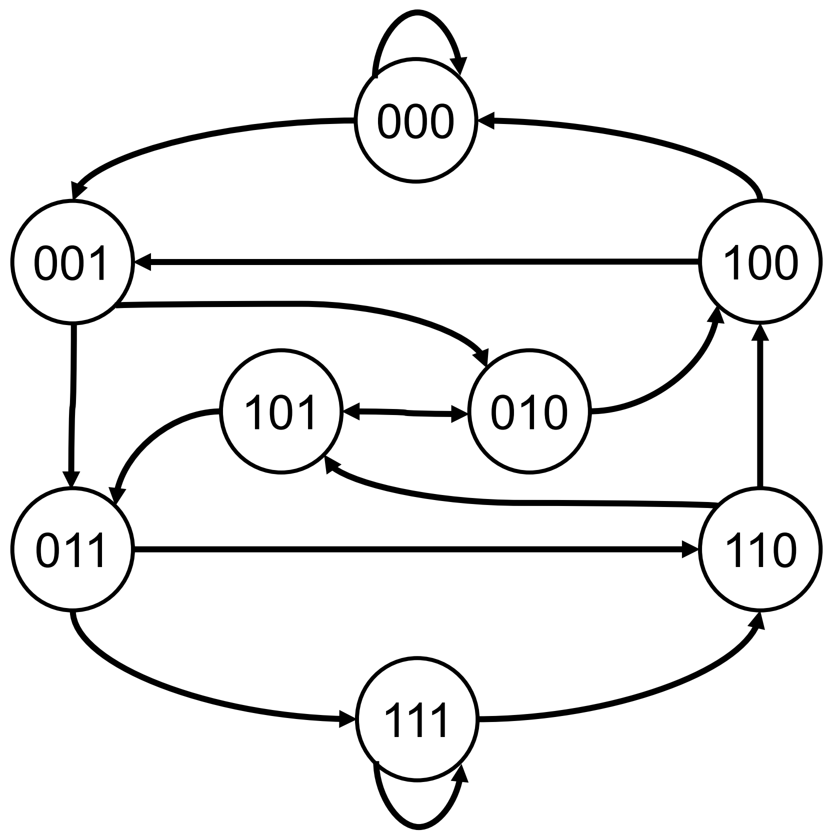
Digit graphs





Digit graphs

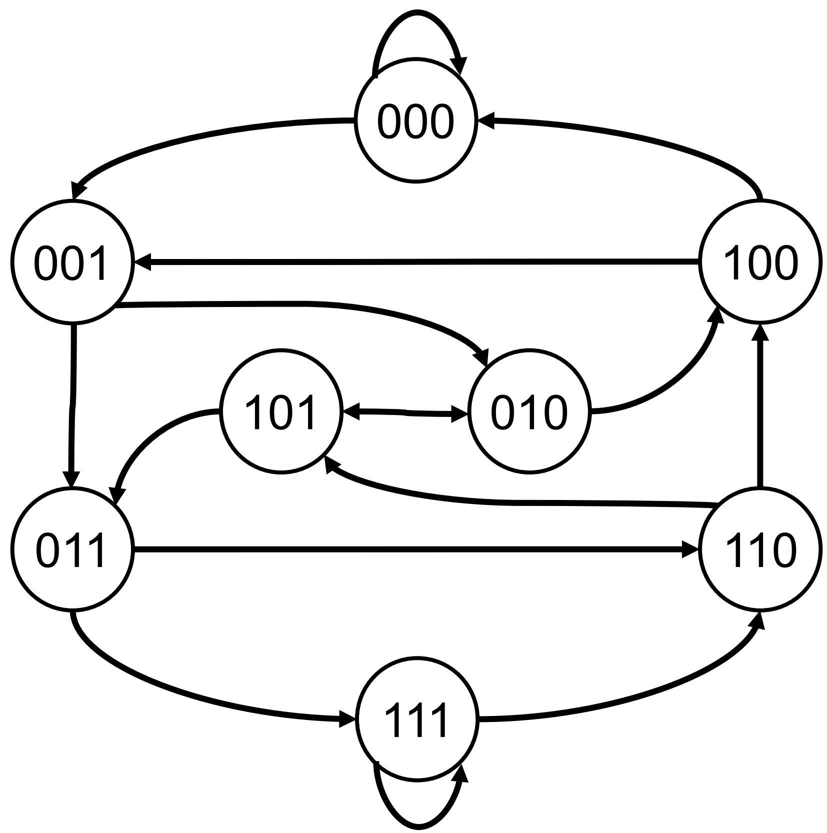
Problem: Find a sequence of maximal length in which no subsequence of length k is repeated.



Digit graphs

Problem: Find a sequence of maximal length in which no subsequence of length k is repeated.

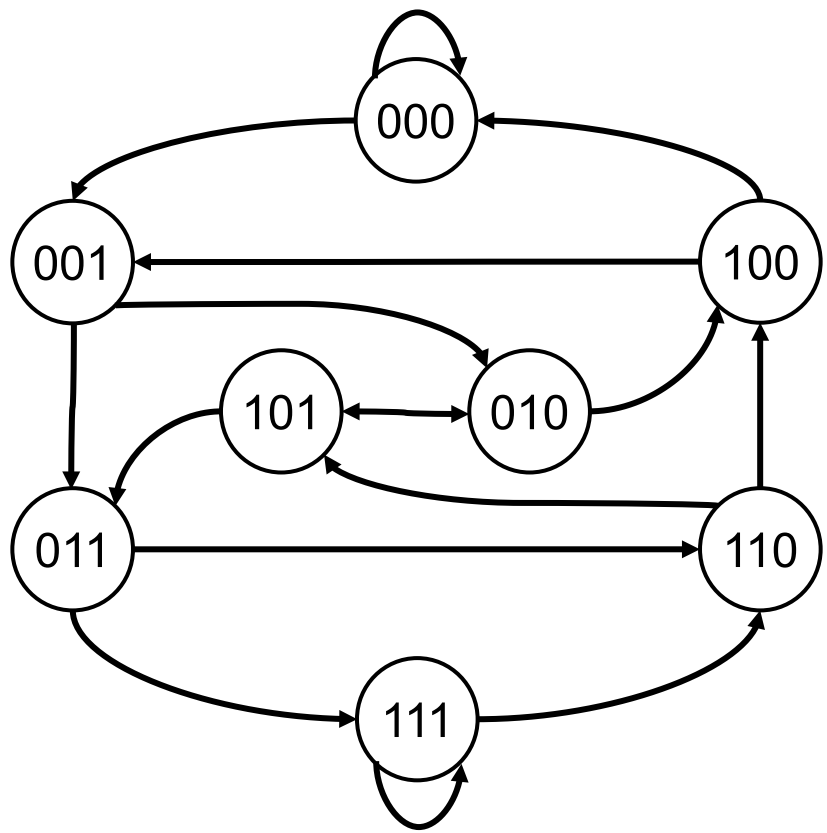
Cannot visit the same vertex twice



Digit graphs

Problem: Find a sequence of maximal length in which no subsequence of length k is repeated.

Visit all vertices



Digit graphs

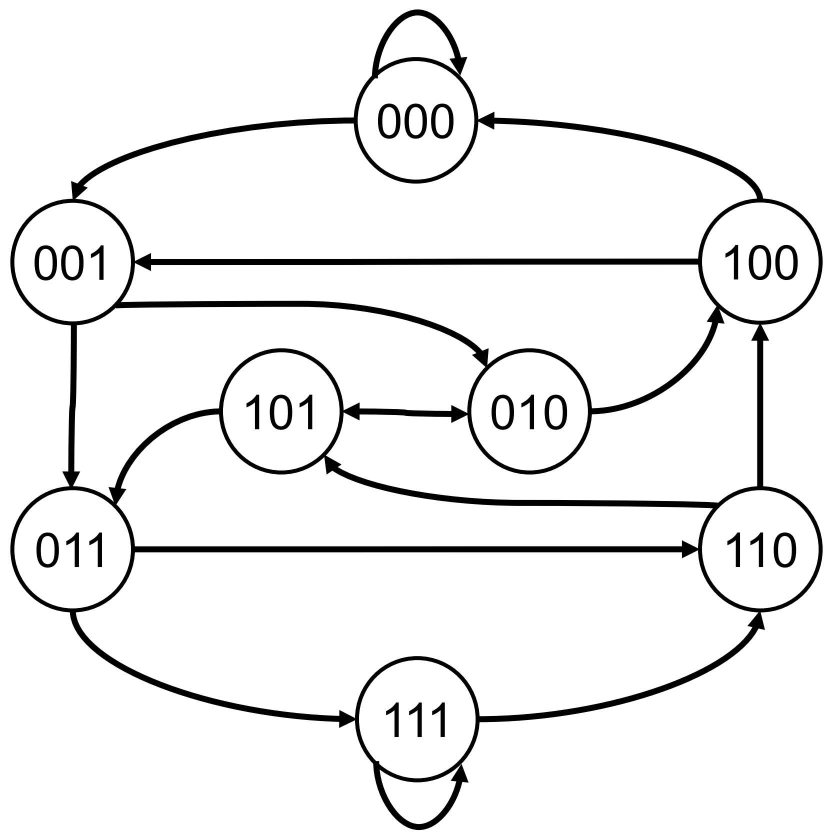
Problem: Find a sequence of maximal length in which no subsequence of length k is repeated.

Find a walk that visits all vertices exactly once.

Sound familiar?

Hamiltonian path!

(Still hard though ☹)



Tuples

Back to the drawing board.

Can we model this differently?

$\boxed{001}10$

Tuples

Back to the drawing board.

Can we model this differently?

Every vertex models the last $k-1$ digits.

$$\boxed{0\,01}\,10$$

Tuples

Back to the drawing board.

Can we model this differently?

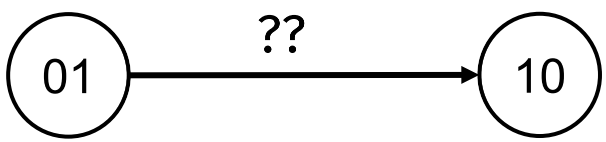
Every vertex models the last $k - 1$ digits.

$$0\boxed{011}0$$

Tuples

Every vertex models the last $k - 1$ digits.

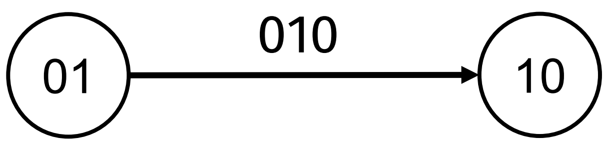
When is there an edge between two vertices?



Tuples

Every vertex models the last $k - 1$ digits.

When is there an edge between two vertices?

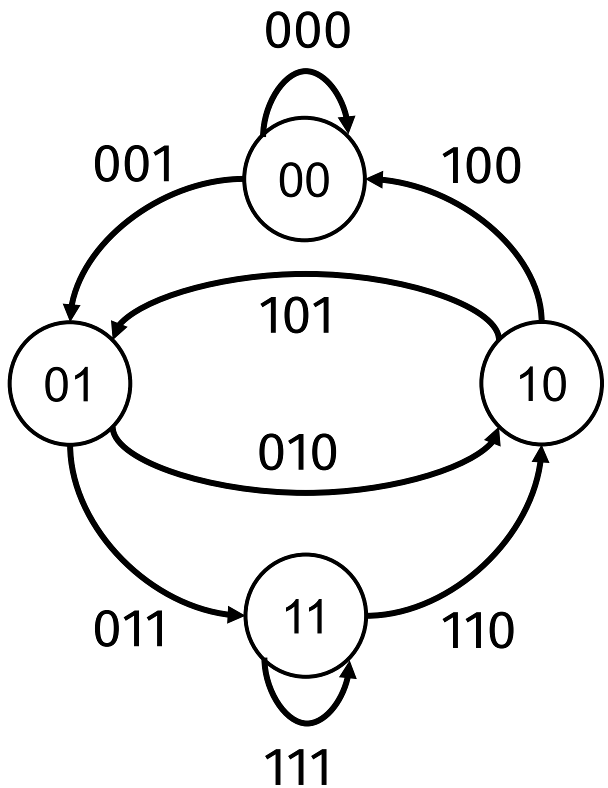


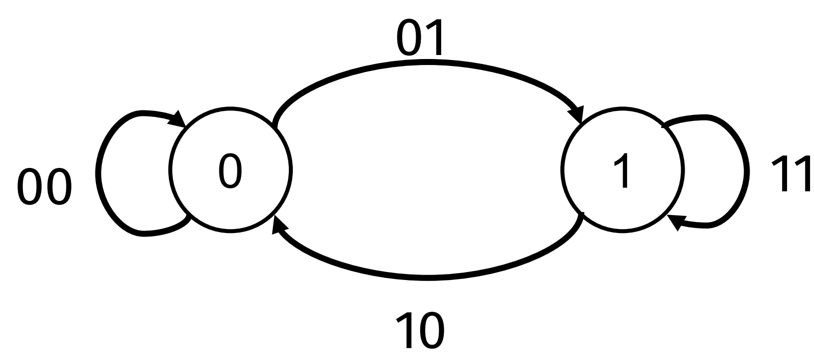
Define a graph $G = (V, E)$ as follows:

- V is the set of all sequences of 0 and 1 of length $k - 1$.
- E is the set of all directed edges between pairs of vertices $((a_1, \dots, a_{k-1}), (a_2, \dots, a_k))$.

(De Bruijn graph)

De Bruijn Graphs

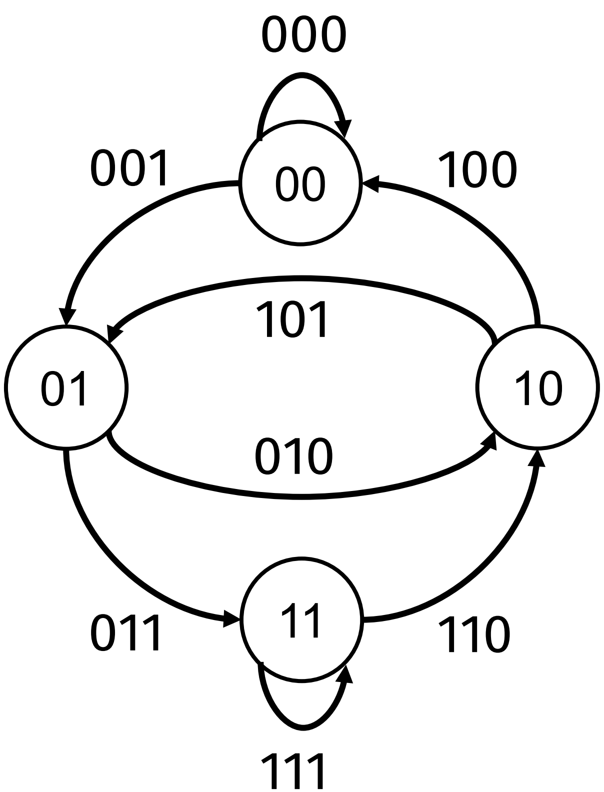




De Bruijn Graphs

Problem: Find a sequence of maximal length in which no subsequence of length k is repeated.

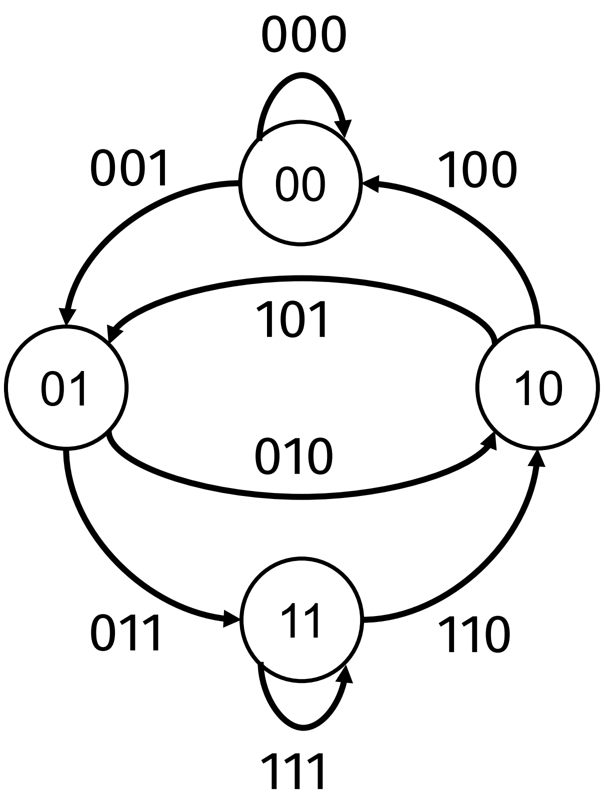
Visit every edge at most once.



De Bruijn Graphs

Problem: Find a sequence of maximal length in which no subsequence of length k is repeated.

Visit every edge.

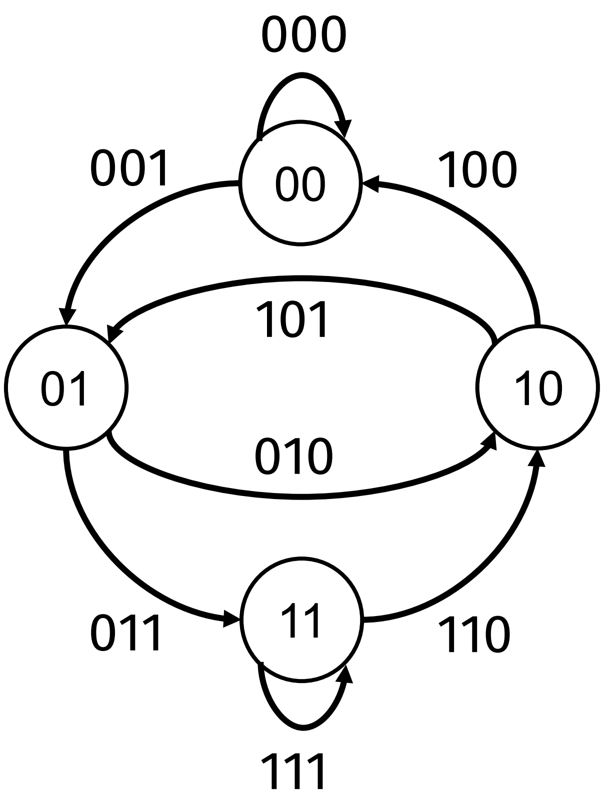


De Bruijn Graphs

Problem: Find a sequence of maximal length in which no subsequence of length k is repeated.

Find a walk that does not visit the same edge twice that visits all edges...

Sounds familiar?



De Bruijn Graphs

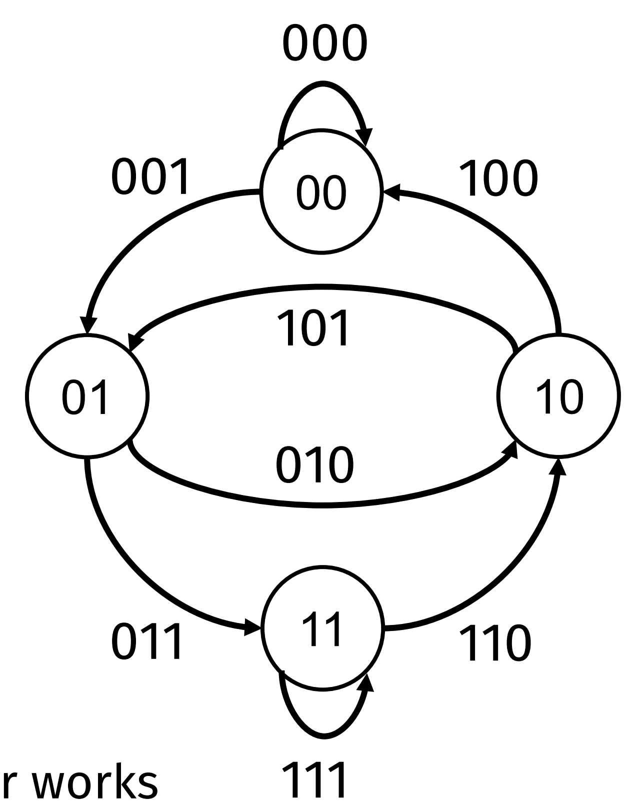
Problem:

Find a cyclic sequence of digits 0 and 1, as long as possible, such that no two k-tuples of consecutive digits coincide.

Find a directed Eulerian tour*.

What are the criteria?

- $\deg^+(v) = \deg^-(v)$
- $sym(G)$ connected



*we do not actually need a tour, but a tour works

De Bruijn Graphs

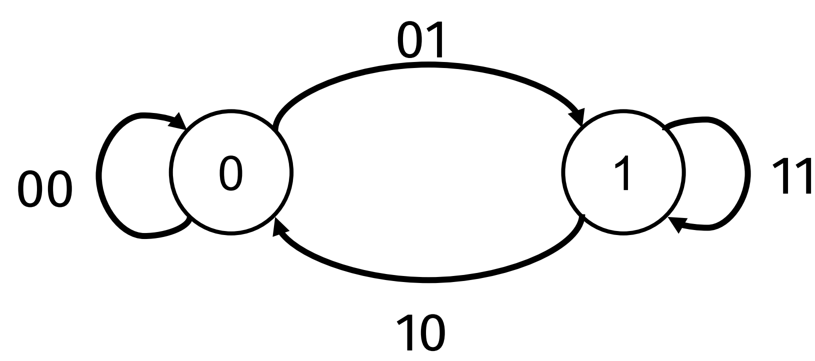
Lemma: Every vertex v in a De Bruijn graph has $\deg^+(v) = \deg^-(v)$

Proof: Let $v = (a_1, \ldots, a_k) \in V$.

By definition of a De Bruijn graph, all outgoing edges from v end in a vertex $(a_2, \ldots, a_k, s)$, where $s \in \{0,1\}$. There are only two of these.

By definition of a De Bruijn graph, all incoming edges to v come from a vertex $(s, a_1, \ldots, a_{k-1})$, where $s \in \{0,1\}$. There are only two of these.

But then $\deg^+(v) = \deg^-(v) = 2$. As v is an arbitrary vertex in the graph it must hold for all vertices in the graph.



De Bruijn Graphs

Lemma: For any De Bruijn graph G, $sym(G)$ connected.

Proof: We prove that there is a walk from every vertex $u = (a_1, \dots, a_k) \in V$ to every other vertex $v = (b_1, \dots, b_k) \in V$.

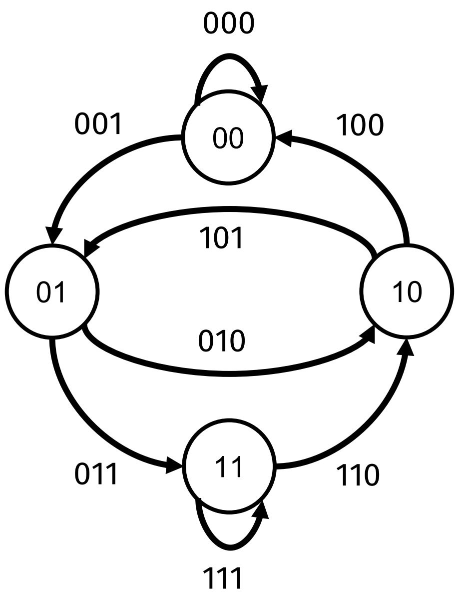
For every vertex $w = (a_p, \dots, a_{p+k-1})$ there exist an outgoing edge of the shape $((a_p, \dots, a_{p+k-1}), (a_{p+1}, \dots, a_{p+k-1}, s))$, for every $s \in \{0,1\}$.

But then starting at u there must be a walk that inserts the consecutive elements of the sequence at v in order at the back of the sequence. After k edges we must reach v.

Thus there must be a walk from every vertex to every other vertex.

$$
\begin{aligned}
&u = (a_1, \dots, a_k) \\
&\to (a_2, \dots, a_k, b_1) \\
&\to (a_3, \dots, a_k, b_1, b_2) \\
&\to \dots \to (a_k, b_1, \dots, b_{k-1}) \\
&\to (b_1, \dots, b_k) = v
\end{aligned}
$$

De Bruijn Graphs



What do we known about these graphs?

- $\deg^+(v) = \deg^-(v) = 2$
- $sym(G)$ connected

- G is a Eulerian directed graph!

So, G admits a Eulerian tour.

Which covers all edges once.

This gives us a sequence of digits of length 2^k.

(take the first digit of each edge travelled)

Keypad Problem

Problem: Find a sequence of maximal length in which no subsequence of length k is repeated.

Starting from an arbitrary vertex in a De Bruijn Graph we create 2^k different sequences in 2^k steps.

Keypad Problem

Problem: Find a sequence of maximal length in which no subsequence of length k is repeated.

Starting from an arbitrary vertex in a De Bruijn Graph we create 2^k different sequences in 2^k steps.

Need to already add the sequence of the starting vertex. So total sequence length is $2^k + k - 1$

The keypad problem

Problem: What is a minimal length of a sequence of 0's and 1's that contains every sequence of length k as a subsequence.

Lower bound:

a sequence of length $2^k + k - 1$ contains exactly 2^k subsequences of length k. So sequence needs at least this length

Upper bound:

Using De Bruijn Graphs we can create a sequence of length $2^k + k - 1$ that contains 2^k different subsequences of length k.

Since there are only 2^k sequences of length k this sequence contains all of them.

Directed acyclic graphs (DAG)

Directed Acyclic Graph (DAG)

A DAG is a directed acyclic graph.

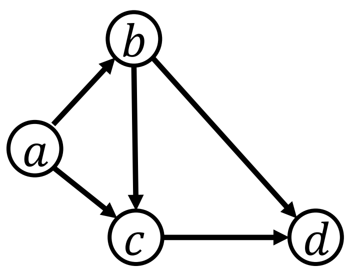
A source is a vertex v, where $\deg^+(v) = 0$
A sink is a vertex v where $\deg^-(v) = 0$.

Example:

G is a DAG

(a) is a source

(d) is a sink



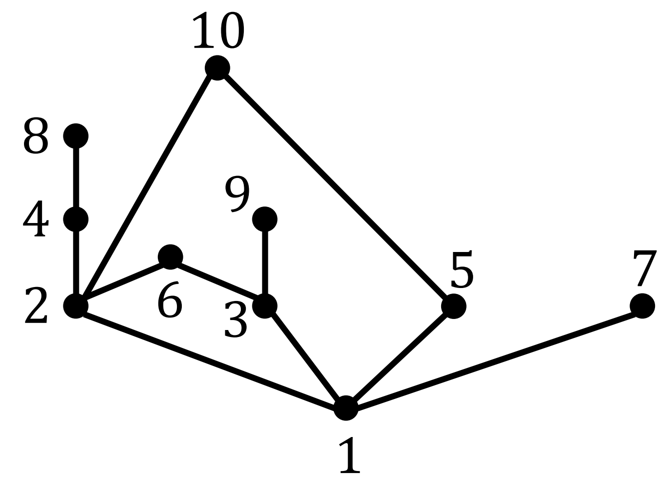
$G = (V, E)$

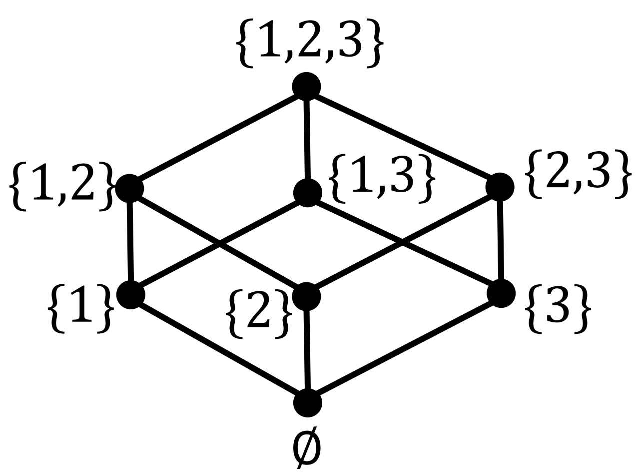
Hasse diagram

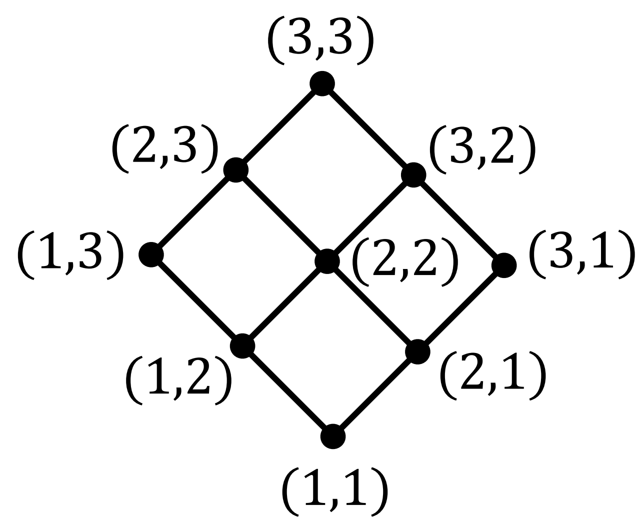
Graph of a partially ordered set $(X, \preccurlyeq)$:

Vertices are elements of the set

There is a directed edge (a, b) if $a \lhd b$
(if a is an immediate predecessor of b)

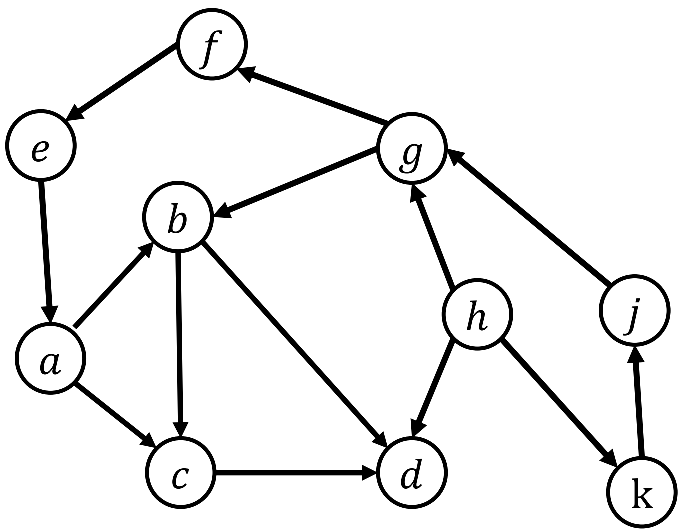






DAG sink

Does every DAG have a sink (source)?



DAG sinks

Theorem: Every (finite) DAG $G = (V, E)$ has at least one sink.

Proof sketch: Take an arbitrary vertex $v \in \mathrm{V}$.

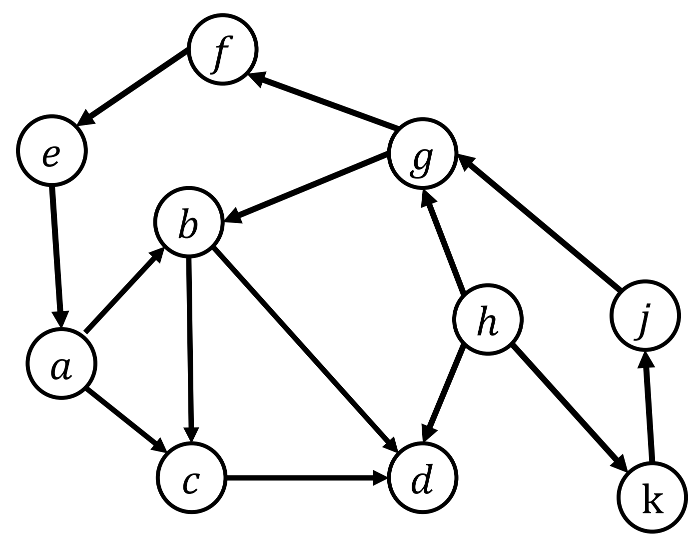
If v is a sink then we are done.

Otherwise v has at least one outgoing edge $e = (v, v')$.

Consider v' and repeat the argument. As G is acyclic we cannot get back to v from v' or from any of the vertices reachable from v'.

So we can visit each vertex only once during this process.

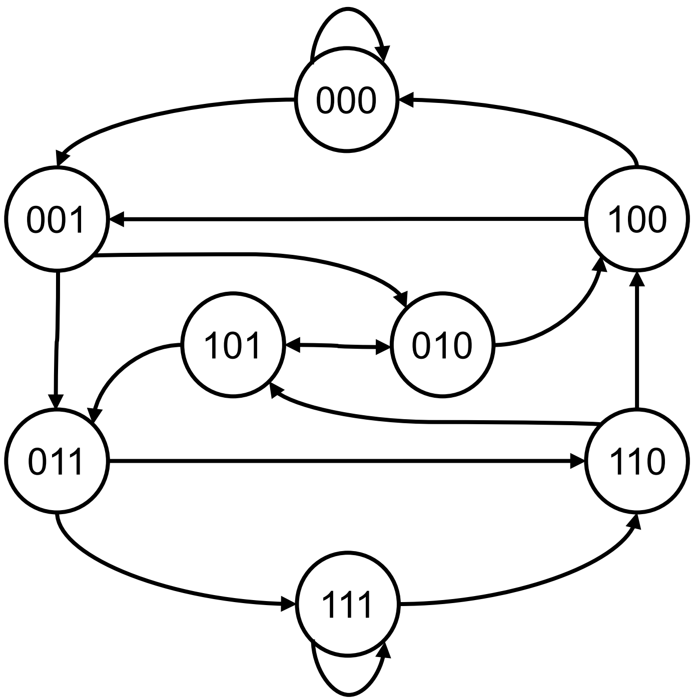
At the latest at the $|V|$-th iteration we must reach a sink.



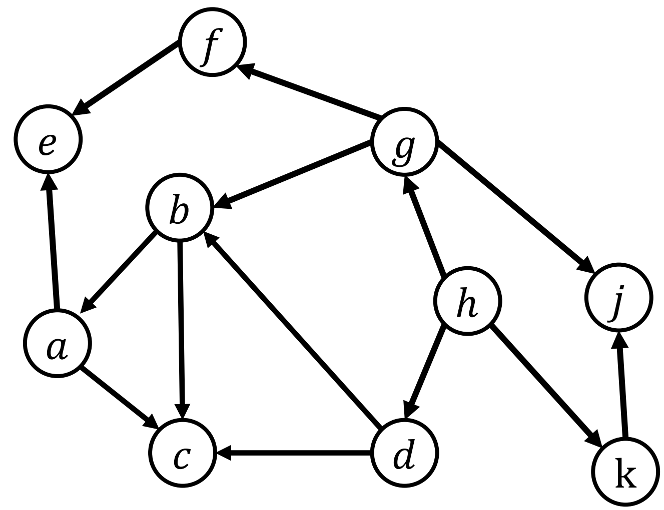
Can make proof formal similar to minimal elements in orderings

Summary

De Bruijn graphs



DAG: Directed Acyclic Graph



Practical stuff

- Practice set
  - Ex. 1 together in discussion group today
  - Ex. 2 for next discussion group

Happy holiday!