Graphs: Trajectories, cycles, connectivity (some material from Rosen, 7th edition)

There are practical problems that can be framed as finding a path in a graph with certain characteristics (shortest path, lowest cost path, etc.)

- Determine whether messages can be exchanged between two computers.
- Route planning for efficient mail delivery.
- Diagnostics in computer networks
- Paths of contagion

Path in an undirected graph

DEFINITION 1 Let n be a nonnegative integer and G an undirected graph. A *path* of *length* n from u to v in G is a sequence of n edges $e_1, \ldots, e_n$ of G for which there exists a sequence $x_0 = u, x_1, \ldots, x_{n-1}, x_n = v$ of vertices such that e_i has, for $i = 1, \ldots, n$, the endpoints x_{i-1} and x_i. When the graph is simple, we denote this path by its vertex sequence $x_0, x_1, \ldots, x_n$ (because listing these vertices uniquely determines the path). The path is a *circuit* if it begins and ends at the same vertex, that is, if $u = v$, and has length greater than zero. The path or circuit is said to *pass through* the vertices $x_1, x_2, \ldots, x_{n-1}$ or *traverse* the edges $e_1, e_2, \ldots, e_n$. A path or circuit is *simple* if it does not contain the same edge more than once.

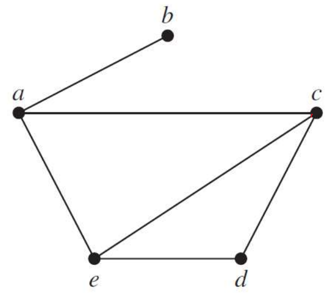


FIGURE 1 **A Simple Graph.**

Path in a directed graph

DEFINITION 2 Let n be a nonnegative integer and G a directed graph. A *path* of length n from u to v in G is a sequence of edges $e_1, e_2, \ldots, e_n$ of G such that e_1 is associated with (x_0, x_1), e_2 is associated with (x_1, x_2), and so on, with e_n associated with (x_{n-1}, x_n), where $x_0 = u$ and $x_n = v$. When there are no multiple edges in the directed graph, this path is denoted by its vertex sequence $x_0, x_1, x_2, \ldots, x_n$. A path of length greater than zero that begins and ends at the same vertex is called a *circuit* or *cycle*. A path or circuit is called *simple* if it does not contain the same edge more than once.

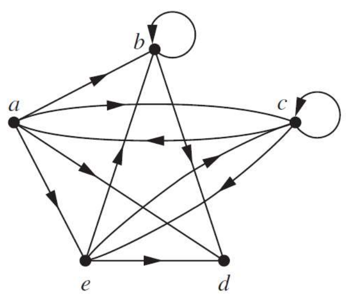


FIGURE 2 A Directed Graph.

Connectivity in undirected graphs

DEFINITION 3 An undirected graph is called *connected* if there is a path between every pair of distinct vertices of the graph. An undirected graph that is not *connected* is called *disconnected*. We say that we *disconnect* a graph when we remove vertices or edges, or both, to produce a disconnected subgraph.

- A **disconnected graph** is the union of two or more connected graphs that do not have vertices in common.
- Each of these subgraphs are the **connected components** of the graph.

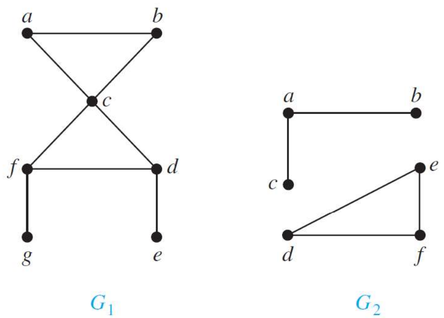


FIGURE 2 **The Graphs G_1 and G_2.**

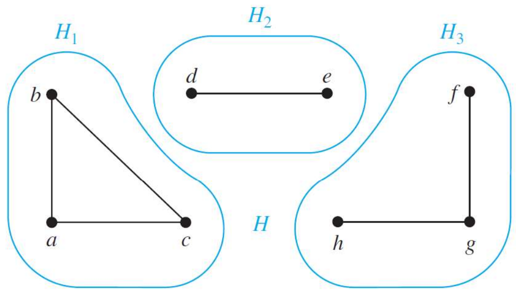


FIGURE 3 **The Graph H and Its Connected Components H_1, H_2, and H_3.**

Simple paths in connected graphs

THEOREM 1 There is a simple path between every pair of distinct vertices of a connected undirected graph.

Proof: Let u and v be two distinct vertices of the connected undirected graph $G = (V, E)$. Because G is connected, there is at least one path between u and v. Let $x_0, x_1, \ldots, x_n$, where $x_0 = u$ and $x_n = v$, be the vertex sequence of a path of least length. This path of least length is simple. To see this, suppose it is not simple. Then $x_i = x_j$ for some i and j with $0 \leq i < j$. This means that there is a path from u to v of shorter length with vertex sequence $x_0, x_1, \ldots, x_{i-1}, x_j, \ldots, x_n$ obtained by deleting the edges corresponding to the vertex sequence $x_i, \ldots, x_{j-1}$. ◁

Theorem: Existence of trajectories

- Let G be a graph (either connected or disconnected) such that it has exactly two vertices of odd degree. There is a trajectory connecting those vertices.

  Proof [By contradiction]:

  - Assume that there is no trajectory between the two vertices. Then each of these two vertices must be in a different connected components of the graph.
  - However, since each of these connected components is a graph on its own, it would mean that such graph has an odd number of vertices (1) with odd degree (1), which is not possible.

Connectivity in directed graphs

DEFINITION 4 A directed graph is *strongly connected* if there is a path from a to b and from b to a whenever a and b are vertices in the graph.

DEFINITION 5 A directed graph is *weakly connected* if there is a path between every two vertices in the underlying undirected graph.

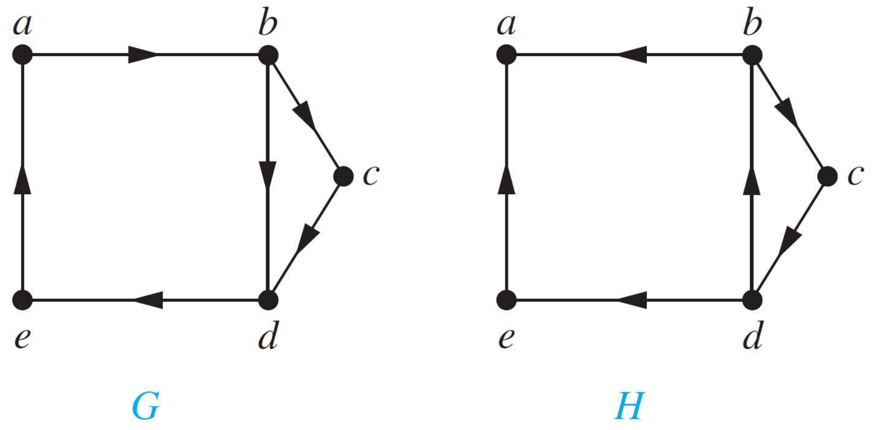


FIGURE 5 **The Directed Graphs G and H.**

Paths and isomorphism

- There is a one-to-one mapping of the paths (trajectories, circuits) defined in two isomorphic graphs.
- The number of simple circuits of length k is invariant under an isomorphism.
- Application: If one graph has a simple circuit of a given length and the other one does not, they cannot be isomorphic.

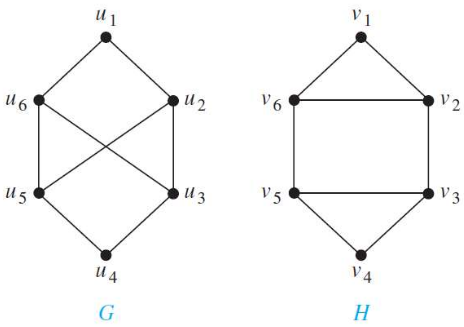


FIGURE 6 The Graphs G and H.

Counting paths

THEOREM 2 Let G be a graph with adjacency matrix $\mathbf{A}$ with respect to the ordering $v_1, v_2, \ldots, v_n$ of the vertices of the graph (with directed or undirected edges, with multiple edges and loops allowed). The number of different paths of length r from v_i to v_j, where r is a positive integer, equals the (i, j)th entry of $\mathbf{A}^r$.

- This theorem is valid even when there are loops and parallel edges.
- Proof [By induction]:
  - For $r = 1$: $A_{ij} =$ Number of paths of length 1 (edges) from node i to node j.
  - Assume that A_{ij}^{r-1} is the number of paths of length $r - 1$ from i to j.

$$A_{ij}^r = \sum_{k=1}^{|V|} A_{ik}^{r-1} A_{kj}$$

# of ways of building a path of length r from i to j by concatenating a path of length $r - 1$ from i to k, with an edge from k to j, for all possible k.

Example: Counting paths

EXAMPLE 15 How many paths of length four are there from a to d in the simple graph G in Figure 8?

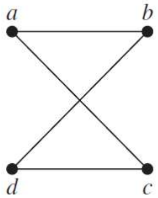


FIGURE 8 The Graph G.

Solution: The adjacency matrix of G (ordering the vertices as a, b, c, d) is

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

Hence, the number of paths of length four from a to d is the $(1, 4)$th entry of $\mathbf{A}^4$. Because

$$\mathbf{A}^4 = \begin{bmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{bmatrix},$$

1. *a, b, a, b, d*
2. *a, b, a, c, d*
3. *a, b, d, b, d*
4. *a, b, d, c, d*
5. *a, c, a, b, d*
6. *a, c, a, c, d*
7. *a, c, d, b, d*
8. *a, c, d, c, d*

Euler’s graph puzzle: The 7 bridges of Köningsberg

- The Seven Bridges of Köningsberg: Is it possible to start at some location in the town, travel across all the bridges once without crossing any bridge twice, and return to the starting point.

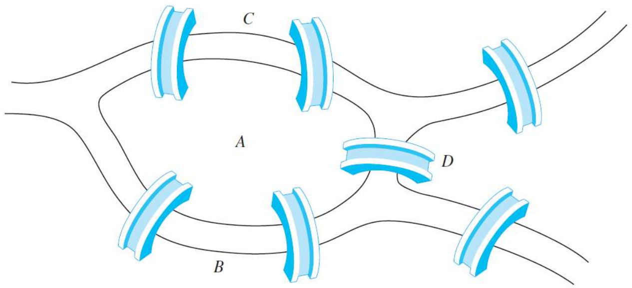


FIGURE 1 The Seven Bridges of Königsberg.

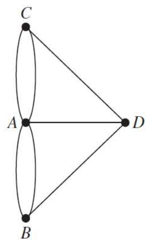
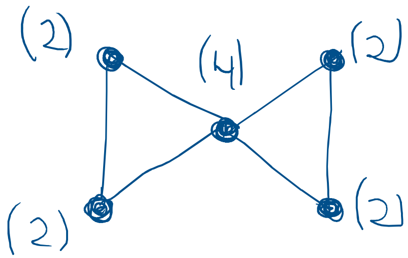


FIGURE 2 Multigraph Model of the Town of Königsberg.

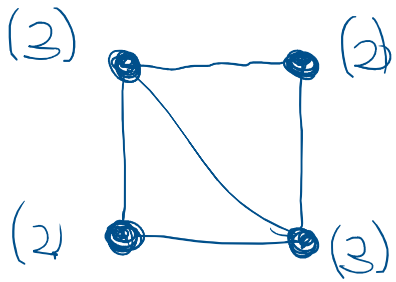
- As a graph problem: Is there a simple circuit in the corresponding multigraph that contains every edge?

Euler circuit, Euler path, Euler graph

- An *Euler circuit* in a graph G is a simple circuit containing every edge of G.
- The graph G is an *Euler graph* if it contains an Euler circuit
- An *Euler path* in G is a simple path containing every edge of G.
- Theorem 1: A connected undirected multigraph G is an *Euler graph* (i.e. has at least one Euler circuit) if and only if all of its vertices have even degree.
- Theorem 2: Graph G has an *Euler path* that is not an *Euler circuit* if exactly two of its vertices has degree odd.



EULER GRAPH



GRAPH WITH AN EULER TRAJECTORY

Example

Can you draw a circuit on this graph without lifting your pen from the paper, so that no part of the picture is retraced?

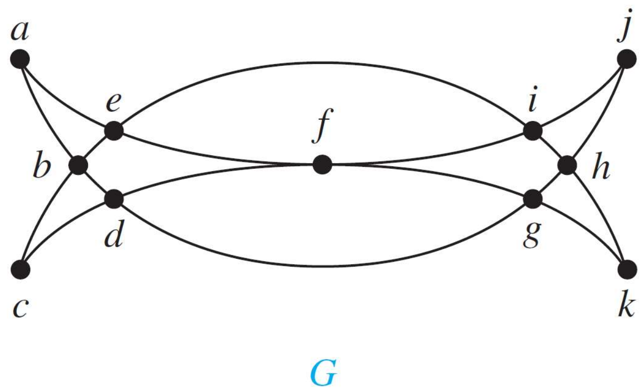


FIGURE 6 **Mohammed's Scimitars.**

Example

Find Euler trajectories, if they exist, in these graphs.

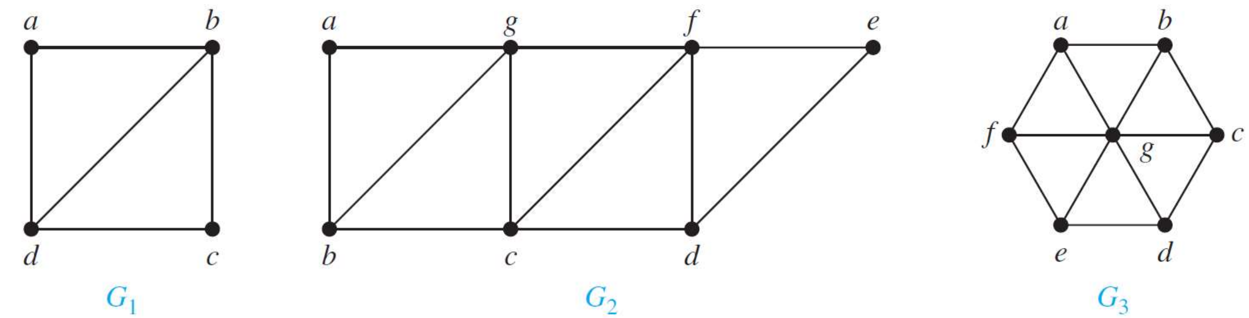


FIGURE 7 **Three Undirected Graphs.**

The 7 bridges of Köningsberg

- Is there a simple circuit in the corresponding multigraph that contains every edge?

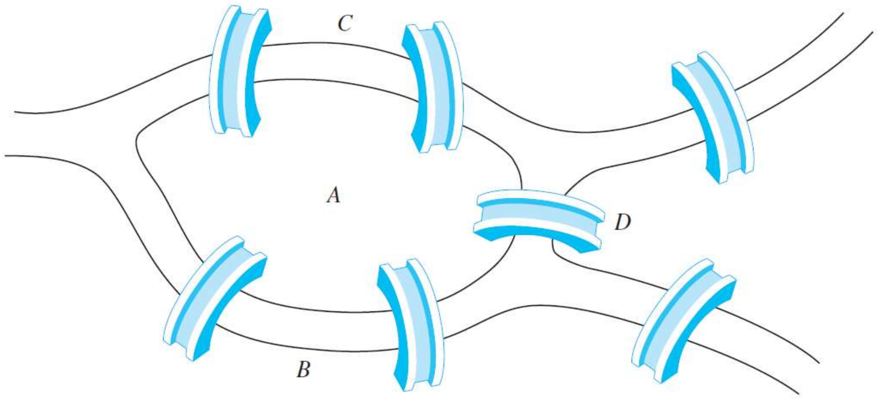


FIGURE 1 **The Seven Bridges of Königsberg.**

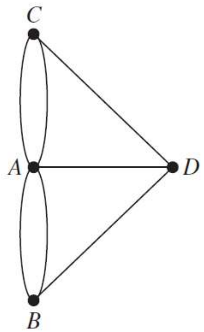
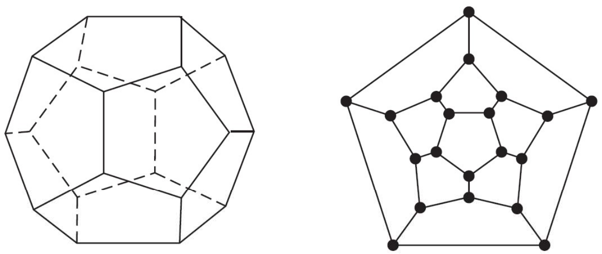


FIGURE 2 **Multigraph Model of the Town of Königsberg.**

Hamilton circuits, Hamilton trajectories

- A *Hamilton circuit* in a graph G is a simple circuit containing every vertex of G exactly once.
- The graph G is a *Hamilton graph* if it contains a *Hamilton circuit.*
- A *Hamilton trajectory* in a graph G is a trajectory that contains every vertex of G exactly once.
- It is possible to build a *Hamilton trajectory* from a *Hamilton circuit* by removing any of the edges of the *Hamilton circuit.*



(a) (b)

FIGURE 8 **Hamilton's "A Voyage Round the World" Puzzle.**

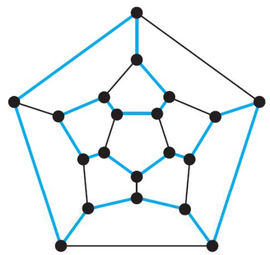


FIGURE 9 **A Solution to the "A Voyage Round the World" Puzzle.**

Graph characteristics and Hamilton circuits

- A complete graph K_n always has a Hamiltonian circuit.
- A graph with a vertex of degree one cannot have a Hamilton circuit, because in a Hamilton circuit, each vertex is incident with two edges in the circuit.
- If a vertex in the graph has degree two, then both edges that are incident with this vertex must be part of any Hamilton circuit.
- When a Hamilton circuit is being constructed and this circuit has passed through a vertex, then all remaining edges incident with this vertex, other than the two used in the circuit, can be removed from consideration.
- A Hamilton circuit cannot contain a smaller circuit within it.

Example

Find Hamilton circuits trajectories, if they exist, in these graphs.

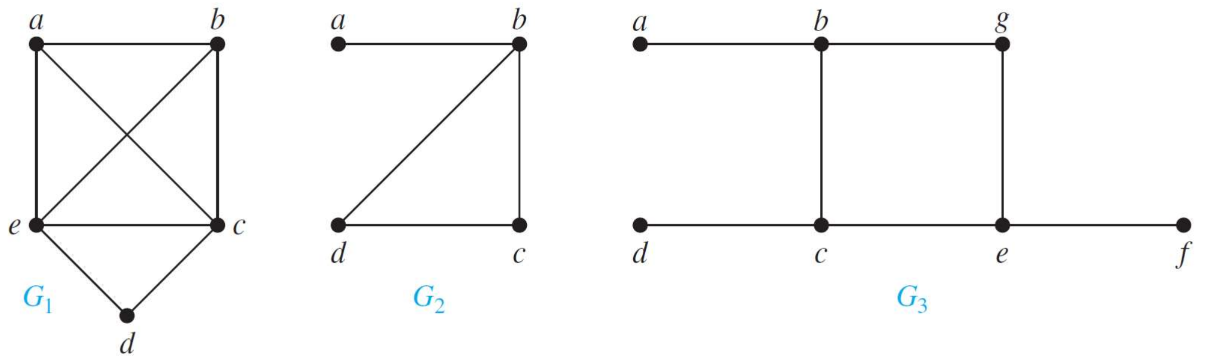


FIGURE 10 **Three Simple Graphs.**

Conditions for existence: Hamilton circuits

- No known necessary and sufficient criteria for the existence of Hamilton circuits.
- However, many theorems are known that give sufficient conditions for the existence of Hamilton circuits.

THEOREM 3 **DIRAC'S THEOREM** If G is a simple graph with n vertices with $n \geq 3$ such that the degree of every vertex in G is at least $n/2$, then G has a Hamilton circuit.

THEOREM 4 **ORE'S THEOREM** If G is a simple graph with n vertices with $n \geq 3$ such that $\deg(u) + \deg(v) \geq n$ for every pair of nonadjacent vertices u and v in G, then G has a Hamilton circuit.

Hamilton circuits: Gray codes

- Gray codes of n bits = Hamilton circuits on Q_n

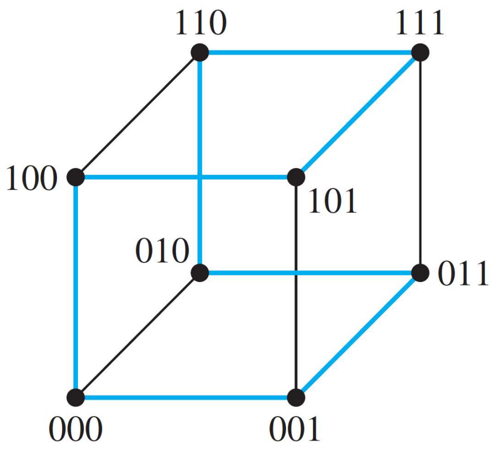
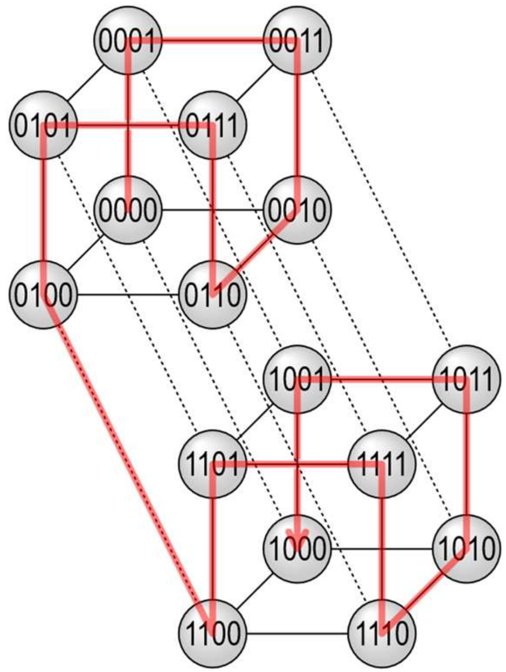
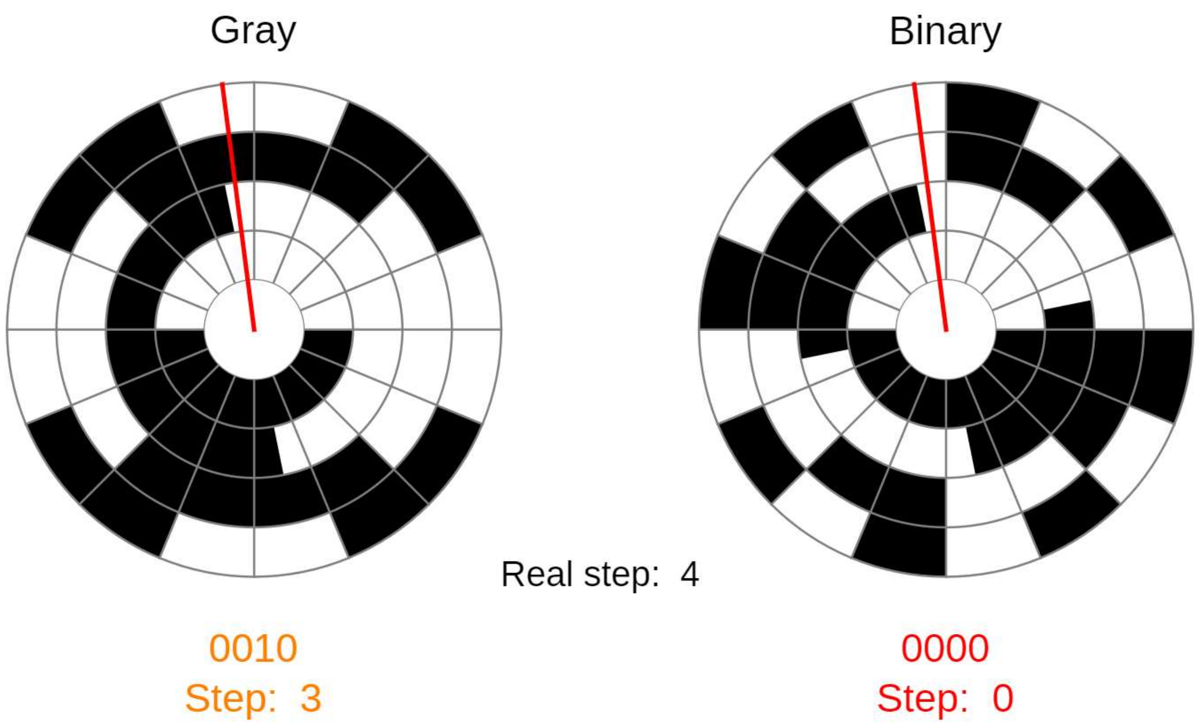


FIGURE 14 **A Hamilton Circuit for Q_3.**



https://commons.wikimedia.org/wiki/File:Gray_code_tesseract.svg

Gray codes: error correction



https://demonstrations.wolfram.com/GrayCodesErrorReductionWithEncoders/

Weighted graphs

- In weighted graph $G = \{V, E\}$, each edge of the graph has a positive weight associated to it $\forall\ e \in E$: $w(e) > 0$.

- Associated problems:
  - Lowest weight path between two vertices: Dijkstra, Warshall
  - Chinese postman's problem: Lowest cost path that guarantees traversing every edge in a graph.
  - Travelling salesman's problem: Lowest cost path that guarantees visiting every vertex in a graph.

Dijkstra's algorithm

ALGORITHM 1 Dijkstra's Algorithm.

procedure *Dijkstra*(G: weighted connected simple graph, with
all weights positive)
{G has vertices $a = v_0, v_1, \ldots, v_n = z$ and lengths $w(v_i, v_j)$
where $w(v_i, v_j) = \infty$ if $\{v_i, v_j\}$ is not an edge in G}
for $i := 1$ **to** n
    $L(v_i) := \infty$
$L(a) := 0$
$S := \emptyset$
{the labels are now initialized so that the label of a is 0 and all
other labels are ∞, and S is the empty set}
while $z \notin S$
    $u :=$ a vertex not in S with $L(u)$ minimal
    $S := S \cup \{u\}$
    for all vertices v not in S
        if $L(u) + w(u, v) < L(v)$ **then** $L(v) := L(u) + w(u, v)$
        {this adds a vertex to S with minimal label and updates the
        labels of vertices not in S}
return $L(z)$ {$L(z) =$ length of a shortest path from a to z}

Floyd's (Warshall) algorithm

ALGORITHM 2 Floyd's Algorithm.

procedure *Floyd*(G: weighted simple graph)
{G has vertices $v_1, v_2, \ldots, v_n$ and weights $w(v_i, v_j)$
with $w(v_i, v_j) = \infty$ if $\{v_i, v_j\}$ is not an edge}
for $i := 1$ **to** n
  for $j := 1$ **to** n
    $d(v_i, v_j) := w(v_i, v_j)$
for $i := 1$ **to** n
  for $j := 1$ **to** n
    for $k := 1$ **to** n
      if $d(v_j, v_i) + d(v_i, v_k) < d(v_j, v_k)$
        then $d(v_j, v_k) := d(v_j, v_i) + d(v_i, v_k)$
return $[d(v_i, v_j)]$ {$d(v_i, v_j)$ is the length of a shortest path between v_i and v_j for $1 \leq i \leq n, 1 \leq j \leq n$}