Mat 3770
Week 4

Spring 2014

Week 4/5 — Student Responsibilities

- Exam 1 is Monday, 2/17
- Reading: Euler Circuits, Hamilton Circuits, Graph Coloring
- Hwk from Tucker: Section 2.1, 2.2
- Hwk from Rosen: Section 9.5
- Attendance **Frostily** Encouraged

Section 2.2: Hamilton Circuits and Paths

- **Hamilton Circuit**: a tour of a graph in which every **vertex** is visited exactly once. We begin and end at the same vertex.
- **Hamilton Path**: a tour of a graph in which ever vertex is visited exactly once, but we do not begin and end at the same vertex.
- **Hamilton** Circuits and Paths are used in routing delivery trucks and in robotic motion planning, say for a drill press that makes holes at predetermined specific locations.
- **Note**: there is no simple way to determine if an arbitrary graph has (or doesn't have) a Hamiltonian Circuit (or HamPath).

Proving Non–existence

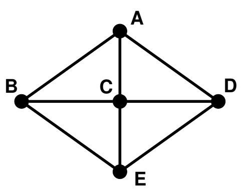
To prove non–existence, begin building parts of a Hamilton Circuit and show **systematically** the construction must fail.

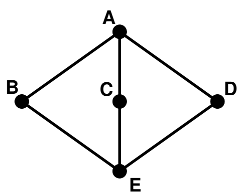
Idea: **any Hamilton Circuit must contain exactly two edges incident to each vertex.**

Three Rules for building a Hamilton Circuit

1. If a vertex v has **degree 2**, both of the edges incident to v must be part of any Hamiltonian Circuit.
2. No proper **subcircuit** (that is, a circuit not containing all vertices) can be formed while constructing a HC.
3. Once the HC is required or forced to use two edges at a vertex v, all other (unused) edges incident to v can be **discarded**.

Building Hamiltonian Circuits





Find a Hamiltonian Circuit if one exists

Theoretical Results

Theorem. A connected graph with $n > 2$ vertices has a Hamilton Circuit if the degree of each vertex is at least $\frac{n}{2}$.

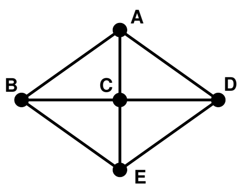
Theorem. Let G be a connected graph with n vertices,
$v_1, v_2, \ldots, v_n \ni \deg(v_i) \leq \deg(v_{i+1}) \ \forall \ 1 \leq i < n$.

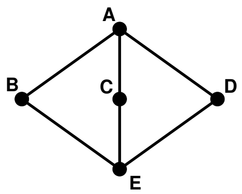
If for each $k \leq \frac{n}{2}$, either

$$\mathbf{deg}(v_k) > k \quad \text{or} \quad \mathbf{deg}(v_{n-k}) \geq n - k,$$

then G has a Hamilton Circuit.

Examples





n =	5				
k =	1	2	3	4	|5
v_k =	A	B	D	E	|C
deg:	3	3	3	3	|4

1	2	3	4	|5
B	C	D	A	|E
2	2	2	3	|3

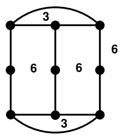
Planar Graphs & HCs

Theorem. Suppose a planar graph G has a Hamilton Circuit H. Let G be drawn with any planar depiction, and

- let r_i denote the number of regions **inside** H bounded by **i edges** in this depiction
- let r_i' be the number of regions **outside** H bounded by **i edges**
- then r_i and r_i' satisfy the equation:

$$\sum_i (i-2)(r_i - r_i') = 0$$

This theorem can be used to show **some** planar graphs cannot have a Ham Circuit.

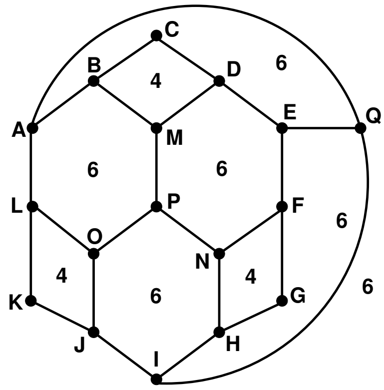


$$r_3 + r_3' = 2$$
$$r_6 + r_6' = 3$$
$$(3-2)(r_3 - r_3') + (6-2)(r_6 - r_6') = 0$$
$$|r_3 - r_3'| \leq 2$$

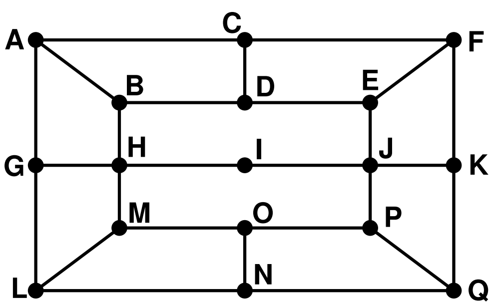
- We cannot have $r_3 - r_3' = 0$ since the equation would then require $r_6 - r_6' = 0$, $\Rightarrow\Leftarrow$ since $r_6 + r_6' = 3$.
- Hence, $(r_6 - r_6') \in \{\pm 1, \pm 3\}$ and so $|4(r_6 - r_6')| \geq 4$
- Now it is impossible to satisfy the equation since $r_3 = 2$, $r_3' = 0$, or vice versa, and $|r_3 - r_3'| = 2$

  Thus, it is impossible for the equation to be valid for this graph, and so no Hamilton Circuit can exist.

Does this graph have a Hamilton Circuit?



Does this graph have a Hamilton Circuit?



Tournaments

Tournament: a directed graph obtained from a complete (undirected) graph, $K_n, n \geq 2$, by giving each edge a direction.

Theorem. Every tournament has a directed **Hamiltonian Path**.

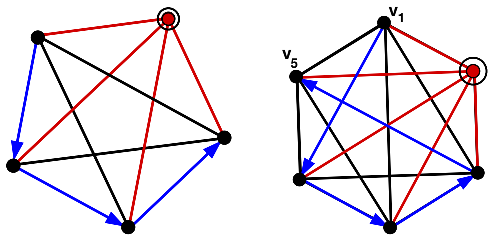
Proof by induction on the number of vertices, n

BC. Let $n = 2$. Then we have two vertices with one edge between them. This edge may be directed toward either of the vertices and trivially we have a directed Hamiltonian Path over K_2.

IH. Assume for some arbitrary $n \geq 2$ that any tournament over K_{n-1} has a directed Hamiltonian Path.

IS. Show any tournament T over K_n has a directed Hamiltonian Path.

Remove an arbitrary vertex x from K_n, leaving a tournament T' over K_{n-1} (with $n-1$ vertices; here we are using the definition of K_n).



By **IH**, T' has a directed HP, say $H = (v_1, \ldots, v_{n-1})$

1. If the edge between x and v_1 is $< x, v_1 >$, then x may be placed at the front of H to obtain a HamPath of T
2. If the edge between x and v_{n-1} is $< v_{n-1}, x >$, then x may be added to the end of H to obtain a HamPath of T
3. Otherwise, we have edges $< v_1, x >$ and $< x, v_{n-1} >$. Then, for some consecutive pair on H, say v_{i-1} and v_i, the edge direction must change (i.e., one goes from path to x, the other from x to path) and thus we can insert x between v_{i-1} and v_i in H and obtain a HamPath of T.

Gray Codes

A scheme to encode information using binary digits

Gray codes are used in transmitting data through space or over the Internet, or for storing information, such as with digital technologies like Compact Disks (CDs) and Digital Video Disks (DVDs).

A satellite transmits images back to Earth. To simplify, let's assume they are black and white with 6 shades of gray, so we need 8 **darkness values** $(1-8)$.

The solution is pretty straight forward—we use 3 bits to encode these values:

1 – 001	2 – 010	3 – 011	4 – 100
5 – 101	6 – 110	7 – 111	8 – 000*

Small Errors = Big Changes

1 – 001	2 – 010	3 – 011	4 – 100
5 – 101	6 – 110	7 – 111	8 – 000*

Notice that if there is an error in sending '3' and a bit gets flipped, we may end up with 111 or 7—a large difference from 3.

Gray Codes attempt to **minimize the effects of errors** – so if one bit gets changed, the result isn't very much different from the true value.

Thus, the scheme is to encode two consecutive decimal numbers by binary sequences that are almost the same – differing in just one position.

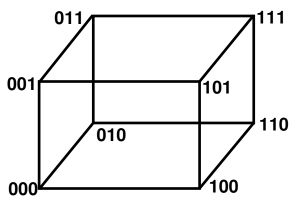
A New Encoding

- If we use 3 bits, then 011 differs by one bit from 001 and 010.
- These can be used for any three number sequence, such as 4 (001), 5 (011), and 6 (010), even though these binary numbers themselves are **not** sequential!
- It is the **mapping** which is important.
- Using a Gray Code doesn't eliminate all errors, but it does cut down on them.
- But what does this have to do with Graph Theory?

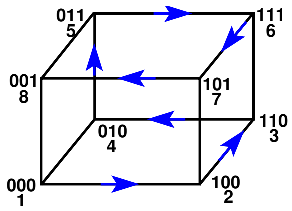
Gray Codes & Hamilton Circuits

We can model the problem of finding a Gray Code (say for the 8 darkness numbers) using a graph and finding a Hamiltonian Circuit.

Each vertex corresponds to a 3–digit binary sequence, and 2 vertices are adjacent if their binary sequences differ by just one bit. This graph turns out to be a cube:



A Hamilton Circuit/Path Mapping



The order of vertices in a Hamilton Path produces a Gray Code since consecutive vertices, representing consecutive decimal numbers, differ in just one position.

Longer Binary Sequences

- For any $n > 0$, similar graphs can be drawn for the 2^n n–digit binary sequences, using an n–dimensional cube or **Hypercube**.
- An n–dimensional cube has 2^n vertices, each of degree n.
- Hypercubes have the property that the longest Hamilton Path between any 2 vertices has length $n - 1$.
- These hypercubes are used in massively parallel computers with 2^n processors

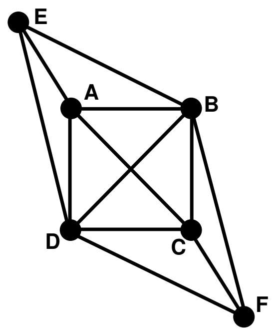
Sec 2.3. Graph Coloring

Recall the four–color map problem from 1.4.

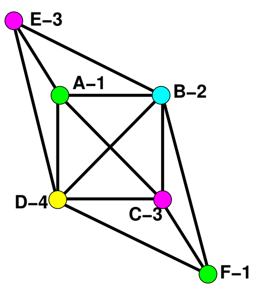
In general, a **coloring** of a graph G assigns colors to the vertices of G so adjacent vertices are given different colors.

Note: vertices with a common color will be mutually non–adjacent. In other words, no same–colored pair of vertices is joined by an edge.

Example Graph



Example Graph Coloring



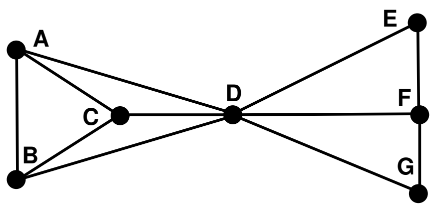
We cannot color with three colors since some adjacent pair in the A–B–C–D subgraph would have the same color.

Note 1. The complete subgraph A–B–C–D **requires** at least 4 colors.

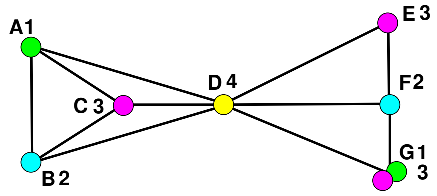
Rule: a complete subgraph on k vertices requires k colors.

Note 2. When building a k–coloring, we can ignore all vertices of degree less than k since when other vertices are colored, there will always be at least one color available to properly color each such vertex.

Another Graph to Color



Example Coloring



Chromatic Number of a Graph

The **chromatic number** of a graph is the **minimal number of colors required to color the graph**.

To **verify** a chromatic number, k, of a graph, we must show:

1. The graph can be colored with k colors.
2. the graph cannot be colored with $k-1$ colors (similar to proving a graph has no Hamilton Circuit, or cannot be isomorphic to another particular graph).

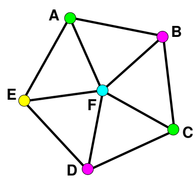
The goal in attempting to prove a chromatic number k is to show any $(k-1)$–coloring forces at least two adjacent vertices to have the same color.

In any coloring, vertices with the same color will be mutually non–adjacent. In other words, they will form an **independent set**.

Wheel Graphs

Wheel graphs are formed from a central vertex with spokes (edges) out to the other vertices, and connections between neighboring outer vertices.

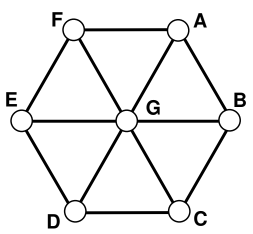
The largest subgraph in a **wheel** graph is a triangle:

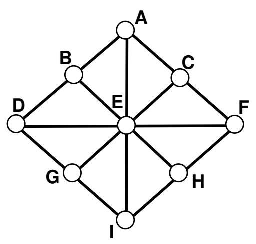


Using colors 1(green), 2(magenta), 3(cyan), 4(yellow), . . . , pick a triangle and assign the first three colors, say to triangle `A–B–F`. This forces `C` to be 1(green), `D` to be 2(magenta), requiring 4(yellow) for `E`.

This wheel cannot be 3–colored.

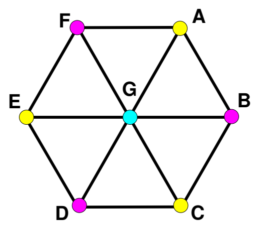
Wheel Coloring Exercise

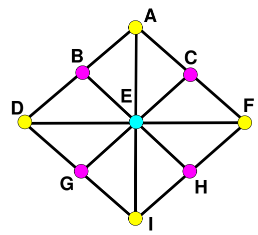




Choose a triangle and find a coloring for each wheel
How many colors are necessary?

Example Wheel Colorings





Why are these wheels 3–colorable, while the one on the earlier slide required 4 colors?

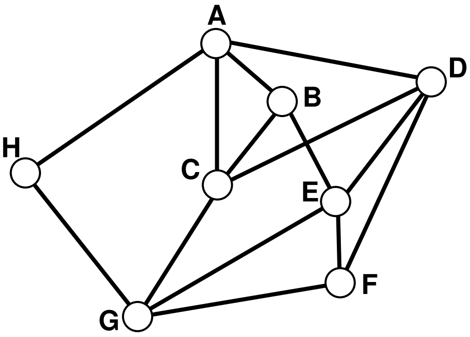
Coloring Wheels

In general, wheels with an even number of spokes are 3–colorable, whereas wheels with an odd number of spokes require 4 colors.

Best bet to find a k–coloring:

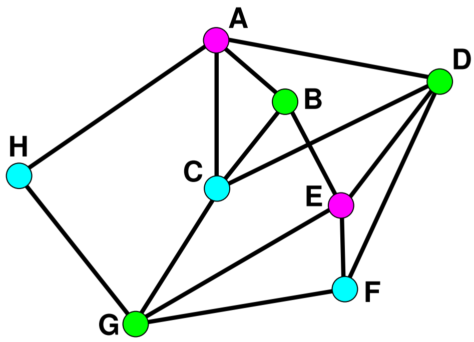
1. start by k–coloring a complete subgraph of k vertices, then
2. find uncolored vertices adjacent to $k-1$ different colored vertices, which forces their color choice.

Generic Coloring Examples—I

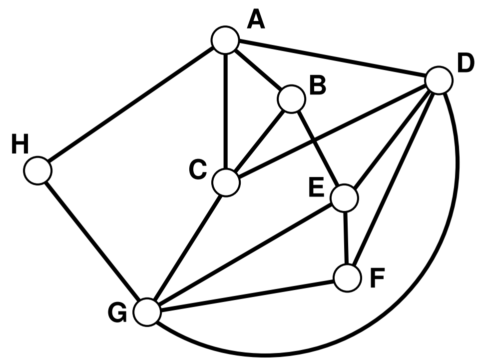


Start with the largest complete subgraph...
What is fewest number of colors needed?

Example Coloring—I

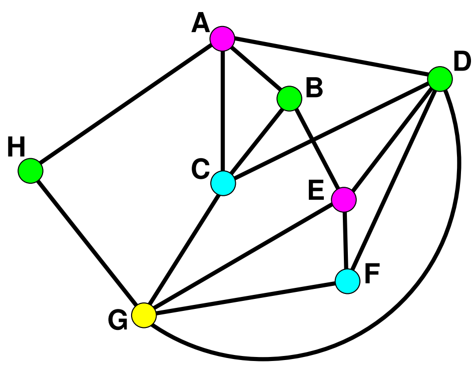


Generic Coloring Examples—II

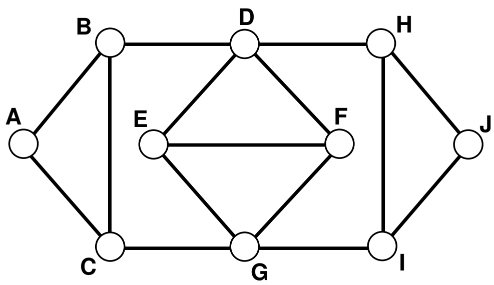


Add one more edge to the previous graph...
What is fewest number of colors needed?

Example Coloring—II

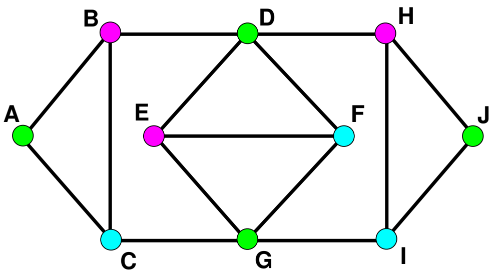


Generic Coloring Examples—III

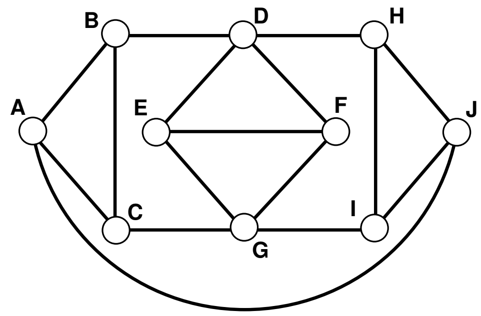


Start with the largest complete subgraph...
What is fewest number of colors needed?

Example Coloring—III

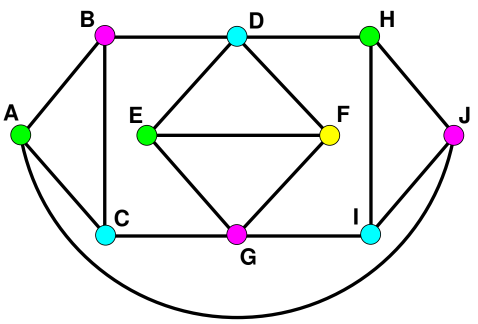


Generic Coloring Examples—IV



Add one more edge to the previous graph. . .
What is fewest number of colors needed?

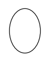
Example Coloring—IV



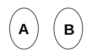
Graph Coloring Applications—Scheduling

Assume we wish to schedule 1–hour meetings for committees which share some members, and we want to minimize the number of meeting hours.

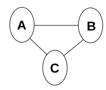
If no committees shared any members, all could meet at the same time.



If no committees shares members with more than 1 other committee, 2 hours would suffice.



Here we would need 3 hours since members may be shared between at most 3 committees.



Graph Coloring and Scheduling

Vertices represent committees (or sports teams, organizations, classes, etc.)

Edges represent "**share one or more members**"

Colors represent disjoint meeting times

This **maps** the scheduling problem to the graph coloring problem. This is an important concept in theoretical and applied computer science.

If we minimize the colors, we minimize the meeting times. . .