Graph Algorithms

Coloring Planar Graphs
**Planar Graphs**

**Definition:** Graphs that can be drawn on the plane with no edge crossing.

⭐ The left figure is a **non-planar** drawing of $K_4$.

⭐ The right figure is a **planar** drawing of $K_4$. 

---

Graph Algorithms
Euler Formula for Planar Graphs

$\star \ m \leq 3n - 6 \text{ for } n \geq 3$: There are not too many edges.
- $n = 1 \Rightarrow m = 0$ and $3n - 6 = -3$.
- $n = 2 \Rightarrow m = 1$ and $3n - 6 = 0$.

$\star$ In $K_5$ there are 5 vertices and 10 edges
$\Rightarrow K_5$ is not planar since $3n - 6 = 9$. 

Graph Algorithms
Euler Formula for Bipartite Planar Graphs

⋆ \( m \leq 2n - 4 \) for \( n \geq 2 \): Bipartite planar graphs have even less edges.
  - \( n = 1 \) \( \Rightarrow \) \( m = 0 \) and \( 2n - 4 = -2 \).

⋆ In \( K_{3,3} \) there are 6 vertices and 9 edges
  \( \Rightarrow \) \( K_{3,3} \) is not planar since \( 2n - 4 = 8 \).
Edge Contraction

**Definition:** Contracting the edge \((u, v)\) in a graph \(G\) is the operation of omitting \(u\) and \(v\) from \(G\) and adding a new vertex \(w\) whose neighbors are all the neighbors of \(u\) and \(v\).

\[
\begin{array}{c}
\text{u} \quad \text{v} \\
\end{array}
\rightarrow
\begin{array}{c}
\text{w} \\
\end{array}
\]

**Lemma:** A planar graph \(G\) remains planar after the contraction of any of its edges.

**Proof:** It is possible to correct the planar drawing of \(G\) to be a planar drawing of the new graph.
Planar Graphs Characterization

**Theorem:** A graph is planar iff it does not contain a “subgraph” $K_5$ or $K_{3,3}$.

**Subgraph:** A graph resulting from the original graph after any sequence of omitting edges and contracting edges.

“Easy”: If a graph contains $K_5$ or $K_{3,3}$ as a subgraph then it is not planar.

“Hard”: The rest of the proof.
The Petersen Graph Contains $K_5$ as a “Subgraph”

Contract the edges $(a, f)$, $(b, g)$, $(c, h)$, $(d, i)$, and $(e, j)$. 
The Petersen Graph Contains $K_{3,3}$ as a “Subgraph”

Omit the edges $(a, e)$ and $(g, i)$
The Petersen Graph Contains $K_{3,3}$ as a “Subgraph”

Contract the edges $(a, b)$ and $(d, e)$
The Petersen Graph Contains $K_{3,3}$ as a "Subgraph"

Contract the edges $(f, i)$ and $(g, j)$
The Petersen Graph Contains $K_{3,3}$ as a “Subgraph”

The final graph is $K_{3,3}$
Lemma: Every planar graph has at least one vertex of degree 5 or less.

Proof:

\[ \sum d(v) = 2m \leq 6n - 12 < 6n. \]

Therefore not all the vertices have degree 6 or more.
Coloring Planar Graphs with 6 Colors

- Omit a vertex of degree 5 or less.
- Color recursively the rest of the graph with 6 colors.
- Color the omitted vertex with a free color.


**Problem:** Find a sequence \( v_1, \ldots, v_n \) such that \( d'(v_i) \leq 5 \) (the back-degree is at most 5).

**Solution:**
- ★ Construct a priority queue of the original degrees.
- ★ Repeat \( n \) times the extract-min and update operations.

**Data Structure:** Adjacency lists to represent the graph.
Complexity

★ $n$ insert operations to construct the initial priority queue.

★ $m = O(n)$ updates and $n$ extract-min operations to find the sequence.

★ Handling the priority queue is $O(n \log n)$ because the complexity of each operation is $O(\log n)$.

★ Finding a free color is $O(1)$ for a total of $O(n)$ to color all the vertices.

★ There is no need to update the adjacency lists.

★ The total complexity is $O(n \log n)$. 
Coloring Planar Graphs with 5 Colors

★ Find a vertex $v$ of degree 5 or less: $d(v) \leq 5$.

★ If $d(v) \leq 4$:
  - Omit $v$ from the graph.
  - Color recursively the rest of the graph with 5 colors.
  - Color $v$ with a free color.
If \( d(v) = 5 \) then since its neighbors are not \( K_5 \), find \( u \) and \( w \) neighbors of \( v \) that are not connected by an edge:

- Contract the edges \((v, u)\) and \((v, w)\) into \( v \).
- Color recursively the new graph with 5 colors.
- Give \( u \) and \( w \) the color of \( v \).
- The neighbors of \( v \) use at most 4 colors.
- Color \( v \) with a free color.
Coloring Planar Graphs with 5 Colors
Complexity

**Data structure:** Adjacency lists for the graph and a priority queue for the degrees.

- Handling the priority queue is $O(n \log n)$ because there are $O(n)$ (= $O(m)$) insert and update and extract-min queue operations each has $O(\log n)$ complexity.

- Combining three adjacency lists into one due to contraction has $O(n)$ complexity for a total of $O(n^2)$ complexity.

- Finding a free color is $O(1)$ for a total of $O(n)$ to color all the vertices.

- The total complexity is $O(n^2)$. 
Another Coloring with 5 Colors

★ Find a vertex $v$ of degree 5 or less: $d(v) \leq 5$.

★ Omit $v$ from the graph.

★ Color recursively the rest of the graph with 5 colors.

★ If $d(v) \leq 4$, then color $v$ with a free color.

★ If the neighbors of $v$ are colored with only 4 colors, then color $v$ with a free color.
Another Coloring with 5 Colors

★ From now on assume that $d(v) = 5$ and that the 5 neighbors $v_1, v_2, v_3, v_4, v_5$ of $v$ ordered clockwise, are colored with the colors $c_1, c_2, c_3, c_4, c_5$ respectively.
Definitions and an Observation

★ For colors $x$ and $y$, let $G(x, y)$ be the subgraph of $G$ containing only the vertices whose colors are $x$ or $y$.

★ For a vertex $w$ whose color is $x$, let $G_w(x, y)$ be the connected component of $G(x, y)$ that contains $w$.

★ Switching between the colors $x$ and $y$ in the connected component $G_w(x, y)$ of $G(x, y)$ results with another legal coloring in which the color of $w$ is $y$. 
The Observation
Another Coloring with 5 Colors

* If $G_{v_1}(c_1, c_3)$ does not contain $v_3$, then switch between the colors $c_1$ and $c_3$ in $G_{v_1}(c_1, c_3)$.

* The color of both $v_1$ and $v_3$ is now $c_3$ and no neighbor of $v$ is colored with $c_1$. Color $v$ with $c_1$. 

Graph Algorithms
Another Coloring with 5 Colors

★ $v_5$ does not belong to $G_{v_2}(c_2, c_5)$ since the graph is planar.
★ Switch between the colors $c_2$ and $c_5$ in $G_{v_2}(c_2, c_5)$.
★ The color of both $v_2$ and $v_5$ is now $c_5$ and no neighbor of $v$ is colored with $c_2$. Color $v$ with $c_2$. 
The problem with an alternative formulation was first posed in 1852.

An incorrect proof that it is possible to color any planar graph with 4 colors appeared in 1879: the error was found 11 years later.

The first correct proof with the help of computers appeared in 1976.

Since then, several researchers simplified the proof but it is still very complicated.
An Incorrect Coloring with 4 Colors

★ Find a vertex $v$ of degree 5 or less: $d(v) \leq 5$.

★ Omit $v$ from the graph.

★ Color recursively the rest of the graph with 4 colors.

★ If the neighbors of $v$ are colored with only 3 colors, then color $v$ with a free color.
There are three cases:

**Case I:** $v$ has 4 neighbors each colored with a different color.

**Case II:** $v$ has 5 neighbors colored with 4 colors. Arranging the neighbors clockwise: the 2 neighbors with the same color are adjacent.

**Case III:** $v$ has 5 neighbors colored with 4 colors. Arranging the neighbors clockwise: the 2 neighbors with the same color are not adjacent.
The Three Cases

Graph Algorithms
Case I

★ Let $v_1, v_2, v_3, v_4$ be the 4 neighbors of $v$ whose colors are $c_1, c_2, c_3, c_4$ respectively.
★ If $G_{v_1}(c_1, c_3)$ does not contain $v_3$, then switch between the colors $c_1$ and $c_3$ in $G_{v_1}(c_1, c_3)$.
★ The color of both $v_1$ and $v_3$ is now $c_3$ and no neighbor of $v$ is colored with $c_1$. Color $v$ with $c_1$. 

Graph Algorithms
Case I

★ Otherwise, $G_{v_1}(c_1, c_3)$ contains $v_3$.
★ $v_4$ does not belong to $G_{v_2}(c_2, c_4)$ since the graph is planar.
★ Switch between the colors $c_2$ and $c_4$ in $G_{v_2}(c_2, c_4)$.
★ The color of both $v_2$ and $v_4$ is now $c_4$ and no neighbor of $v$ is colored with $c_2$. Color $v$ with $c_2$. 
Case II

★ Let $v_1, v_2, v_3, v_4, v_5$ be the 5 neighbors of $v$ whose colors are $c_1, c_2, c_3, c_4, c_4$ respectively.

★ Join $v_4, v_5$ into a new vertex $w$ whose color is $c_4$ and let the neighbors of $w$ be all the neighbors of $v_4, v_5$. The new graph is planar since $v_4, v_5$ are clockwise neighbors of $v$.

★ Color $v$ as in Case I and separate $v_4, v_5$. 

![Graph Algorithms Diagram]
Case III

★ Let \( v_1, v_2, v_3, v_4, v_5 \) be the 5 neighbors of \( v \) whose colors are \( c_1, c_2, c_3, c_4, c_2 \) respectively.

★ If \( G_{v_1}(c_1, c_3) \) does not contain \( v_3 \), then switch between the colors \( c_1 \) and \( c_3 \) in \( G_{v_1}(c_1, c_3) \).

★ The color of both \( v_1 \) and \( v_3 \) is now \( c_3 \) and no neighbor of \( v \) is colored with \( c_1 \). Color \( v \) with \( c_1 \).
Case III

★ Otherwise, $G_{v_1}(c_1, c_3)$ contains $v_3$.

★ If $G_{v_1}(c_1, c_4)$ does not contain $v_4$, then switch between the colors $c_1$ and $c_3$ in $G_{v_1}(c_1, c_3)$.

★ The color of both $v_1$ and $v_4$ is now $c_4$ and no neighbor of $v$ is colored with $c_1$. Color $v$ with $c_1$. 

![Diagram showing the process of coloring vertices in a graph with different colors, transitions, and resulting configurations.](image)
Case III

- Otherwise, $G_{v_1}(c_1, c_4)$ contains $v_4$. Recall that $G_{v_1}(c_1, c_3)$ contains $v_3$.

- $v_4$ does not belong to $G_{v_2}(c_2, c_4)$ and $v_3$ does not belong to $G_{v_5}(c_2, c_3)$ since the graph is planar.
Case III

★ Switch between the colors $c_2$ and $c_4$ in $G_{v_2}(c_2, c_4)$ and switch between the colors $c_2$ and $c_3$ in $G_{v_5}(c_2, c_3)$.

★ The color of both $v_2$ and $v_4$ is now $c_4$ and the color of both $v_5$ and $v_3$ is now $c_3$ and no neighbor of $v$ is colored with $c_2$. Color $v$ with $c_2$. 

Graph Algorithms
The Proof is Incorrect!

After switching between the colors \( c_2 \) and \( c_4 \) in \( G_{v_2}(c_2, c_4) \) and switching between the colors \( c_2 \) and \( c_3 \) in \( G_{v_5}(c_2, c_3) \), there could be 2 neighboring vertices whose color is \( c_2 \).