CS 580: Algorithm Design and Analysis

Jeremiah Blocki
Purdue University
Spring 2019

Homework 5. Due on Thursday, March 28th at 11:59 PM (on Gradescope)

Midterm 2. April 3 @ 8PM (EE 170)
Practice Midterm Released Soon
3x5 Index Card (Double Sided)

Midterm 2

- When?
  - April 3rd from 8PM to 10PM (2 hours)
- Where?
  - EE 170
- What can I bring?
  - 3x5 inch index card with your notes (double sided)
  - No electronics (phones, computers, calculators etc…)

2

Midterm 2

- When?
  - April 3rd from 8PM to 10PM (2 hours)
- Where?
  - EE 170
- What material should I study?
  - The midterm will cover recent topics more heavily
    - Network Flow
      - Max-Flow Min-Cut, Augmenting Paths, etc…
      - Ford Fulkerson, Dinic's Algorithm etc…
      - Applications of Network Flow (e.g., Maximum Bipartite Matching)
    - Linear Programming
    - NP-Completeness
      - Polynomial time reductions, P, NP, NP-Hard, NP-Completess, coNP
    - PSPACE (only basic questions)

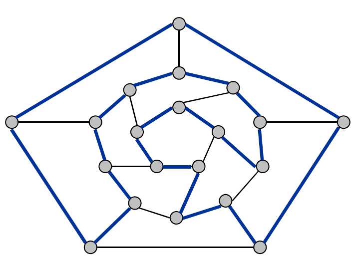
3

8.5 Sequencing Problems

Basic genres.
- Packing problems: SET-PACKING, INDEPENDENT SET.
- Covering problems: SET-COVER, VERTEX-COVER.
- Constraint satisfaction problems: SAT, 3-SAT.
- Sequencing problems: HAMILTONIAN-CYCLE, TSP.
- Partitioning problems: 3D-MATCHING, 3-COLOR.
- Numerical problems: SUBSET-SUM, KNAPSACK.

Hamiltonian Cycle

HAM-CYCLE: given an undirected graph G = (V, E), does there exist a simple cycle Γ that contains every node in V.

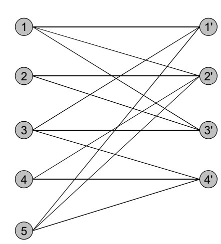


YES: vertices and faces of a dodecahedron.

5

Hamiltonian Cycle

HAM-CYCLE: given an undirected graph G = (V, E), does there exist a simple cycle Γ that contains every node in V.



NO: bipartite graph with odd number of nodes.

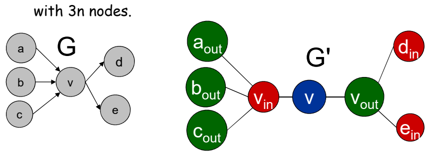
6

Directed Hamiltonian Cycle

DIR-HAM-CYCLE: given a digraph $G = (V, E)$, does there exists a simple directed cycle Γ that contains every node in V?

Claim. DIR-HAM-CYCLE $\le_P$ HAM-CYCLE.

Pf. Given a directed graph $G = (V, E)$, construct an undirected graph G' with 3n nodes.



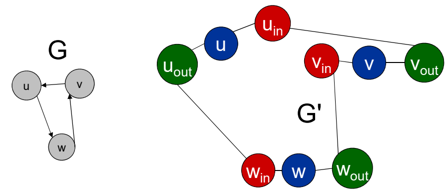
7

Directed Hamiltonian Cycle

Claim. G has a Hamiltonian cycle iff G' does.

Pf. $\Rightarrow$

- Suppose G has a directed Hamiltonian cycle Γ (e.g., (u,w,v)).
- Then G' has an undirected Hamiltonian cycle (same order).
  - For each node v in directed path cycle replace v with v_{in},v,v_{out}



8

Directed Hamiltonian Cycle

Claim. G has a Hamiltonian cycle iff G' does.

Pf. $\Rightarrow$

- Suppose G has a directed Hamiltonian cycle Γ.
- Then G' has an undirected Hamiltonian cycle (same order).
  - For each node v in directed path cycle replace v with v_{in},v,v_{out}

Pf. $\Leftarrow$

- Suppose G' has an undirected Hamiltonian cycle Γ'.
- Γ' must visit nodes in G' using one of following two orders:
  - …, B, G, R, B, G, R, B, G, R, B, …
  - …, B, R, G, B, R, G, B, R, G, B, …
- Blue nodes in Γ' make up directed Hamiltonian cycle Γ in G, or reverse of one. ▪

9

3-SAT Reduces to Directed Hamiltonian Cycle

Claim. 3-SAT $\le_P$ DIR-HAM-CYCLE.

Pf. Given an instance Φ of 3-SAT, we construct an instance of DIR-HAM-CYCLE that has a Hamiltonian cycle iff Φ is satisfiable.

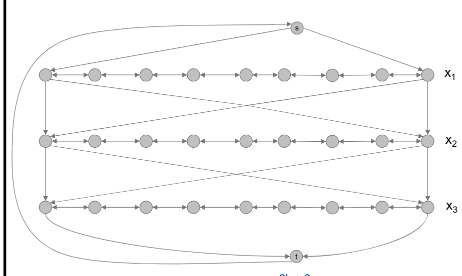
Construction. First, create graph that has 2^n Hamiltonian cycles which correspond in a natural way to 2^n possible truth assignments.

10

3-SAT Reduces to Directed Hamiltonian Cycle

Construction. Given 3-SAT instance Φ with n variables x_i and k clauses.

- Construct G to have 2^n Hamiltonian cycles.
- Intuition: traverse path i from left to right $\Leftrightarrow$ set variable $x_i = 1$.

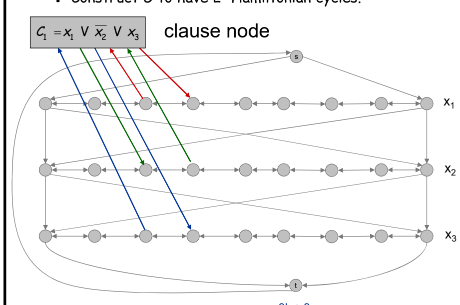


11

3-SAT Reduces to Directed Hamiltonian Cycle

Construction. Given 3-SAT instance Φ with n variables x_i and k clauses.

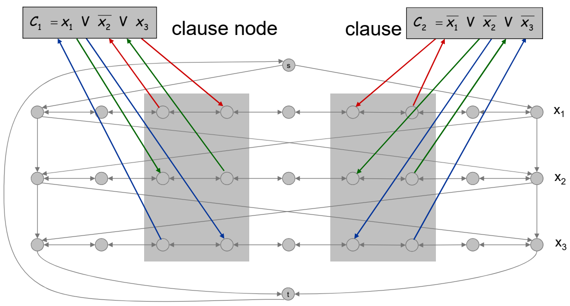
- Construct G to have 2^n Hamiltonian cycles.



12

3-SAT Reduces to Directed Hamiltonian Cycle

Construction. Given 3-SAT instance Φ with n variables x_i and k clauses.
- For each clause: add a node and 6 edges.



3-SAT Reduces to Directed Hamiltonian Cycle

Claim. Φ is satisfiable iff G has a Hamiltonian cycle.

Pf. $\Rightarrow$
- Suppose 3-SAT instance has satisfying assignment x^*.
- Then, define Hamiltonian cycle in G as follows:
  - if $x^*_i = 1$, traverse row i from left to right
  - if $x^*_i = 0$, traverse row i from right to left
  - for each clause C_j, there will be at least one row i in which we are going in "correct" direction to splice node C_j into tour

3-SAT Reduces to Directed Hamiltonian Cycle

Claim. Φ is satisfiable iff G has a Hamiltonian cycle.

Pf. $\Leftarrow$
- Suppose G has a Hamiltonian cycle Γ.
- If Γ enters clause node C_j, it must depart on mate edge.
  - thus, nodes immediately before and after C_j are connected by an edge e in G
  - removing C_j from cycle, and replacing it with edge e yields Hamiltonian cycle on $G - \{ C_j \}$
- Continuing in this way, we are left with Hamiltonian cycle Γ' in $G - \{ C_1, C_2, \ldots, C_k \}$.
- Set $x^*_i = 1$ iff Γ' traverses row i left to right.
- Since Γ visits each clause node C_j, at least one of the paths is traversed in "correct" direction, and each clause is satisfied. ▪

Longest Path

SHORTEST-PATH. Given a digraph $G = (V, E)$, does there exists a simple path of length at most k edges?

LONGEST-PATH. Given a digraph $G = (V, E)$, does there exists a simple path of length at least k edges?

Claim. 3-SAT $\le_P$ LONGEST-PATH.

Pf 1. Redo proof for DIR-HAM-CYCLE, ignoring back-edge from t to s.
Pf 2. Show HAM-CYCLE $\le_P$ LONGEST-PATH.

Traveling Salesperson Problem

TSP. Given a set of n cities and a pairwise distance function d(u, v), is there a tour of length $\le$ D?

All 13,509 cities in US with a population of at least 500
Reference: http://www.tsp.gatech.edu

Traveling Salesperson Problem

TSP. Given a set of n cities and a pairwise distance function d(u, v), is there a tour of length $\le$ D?

Optimal TSP tour
Reference: http://www.tsp.gatech.edu

Traveling Salesperson Problem

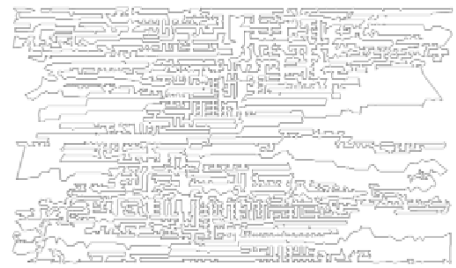
TSP. Given a set of n cities and a pairwise distance function d(u, v), is there a tour of length $\leq$ D?

11,849 holes to drill in a programmed logic array
Reference: http://www.tsp.gatech.edu

19

Traveling Salesperson Problem

TSP. Given a set of n cities and a pairwise distance function d(u, v), is there a tour of length $\leq$ D?



Optimal TSP tour
Reference: http://www.tsp.gatech.edu

20

Traveling Salesperson Problem

TSP. Given a set of n cities and a pairwise distance function d(u, v), is there a tour of length $\leq$ D?

HAM-CYCLE: given a graph G = (V, E), does there exists a simple cycle that contains every node in V?

Claim. HAM-CYCLE $\leq_P$ TSP.
Pf.

- Given instance G = (V, E) of HAM-CYCLE, create n cities with distance function

$$d(u, v) = \begin{cases} 1 & \text{if } (u, v) \in E \\ 2 & \text{if } (u, v) \notin E \end{cases}$$

- TSP instance has tour of length $\leq$ n iff G is Hamiltonian. ▪

Remark. TSP instance in reduction satisfies Δ-inequality.

21

Numerical Problems

Basic genres.

- Packing problems: SET-PACKING, INDEPENDENT SET.
- Covering problems: SET-COVER, VERTEX-COVER.
- Constraint satisfaction problems: SAT, 3-SAT.
- Sequencing problems: HAMILTONIAN-CYCLE, TSP.
- Partitioning problems: 3D-MATCHING, 3-COLOR.
- Numerical problems: SUBSET-SUM, KNAPSACK.

Subset Sum (proof from book)

Construction. Let $X \cup Y \cup Z$ be a instance of 3D-MATCHING with triplet set T. Let n = |X| = |Y| = |Z| and m = |T|.

- Let $X = \{ x_1, x_2, x_3\ x_4\}$, $Y = \{ y_1, y_2, y_3, y_4\}$, $Z = \{ z_1, z_2, z_3, z_4\}$
- For each triplet $t = (x_i, y_j, z_k) \in T$, create an integer w_t with 3n digits that has a 1 in positions i, n+j, and 2n+k.

use base m+1

Claim. 3D-matching iff some subset sums to W = 111,…, 111.

Triplet t_i			x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4	z_1	z_2	z_3	z_4	w_i
x_1	y_2	z_3	1	0	0	0	0	1	0	0	0	0	1	0	100,001,000,010
x_2	y_4	z_2	0	1	0	0	0	0	0	1	0	1	0	0	10,000,010,100
x_1	y_1	z_1	1	0	0	0	1	0	0	0	1	0	0	0	100,010,001,000
x_2	y_2	z_4	0	1	0	0	0	1	0	0	0	0	0	1	10,001,000,001
x_4	y_3	z_4	0	0	0	1	0	0	1	0	0	0	0	1	100,100,001
x_3	y_1	z_2	0	0	1	0	1	0	0	0	0	1	0	0	1,010,000,100
x_3	y_1	z_3	0	0	1	0	1	0	0	0	0	0	1	0	1,010,000,010
x_3	y_1	z_1	0	0	1	0	1	0	0	0	1	0	0	0	1,010,001,000
x_4	y_4	z_4	0	0	0	1	0	0	0	1	0	0	0	1	100,010,001
															111,111,111,111

23

Partition

SUBSET-SUM. Given natural numbers $w_1, \ldots, w_n$ and an integer W, is there a subset that adds up to exactly W?

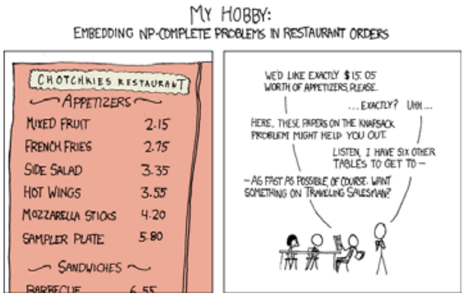
PARTITION. Given natural numbers $v_1, \ldots, v_m$, can they be partitioned into two subsets that add up to the same value?

$\frac{1}{2} \Sigma_i v_i$

Claim. SUBSET-SUM $\leq_P$ PARTITION.
Pf. Let $W, w_1, \ldots, w_n$ be an instance of SUBSET-SUM.

- Create instance of PARTITION with m = n+2 elements.
  - $v_1 = w_1, v_2 = w_2, \ldots, v_n = w_n$, $v_{n+1} = 2\ \Sigma_i w_i - W$, $v_{n+2} = \Sigma_i w_i + W$
- There exists a subset that sums to W iff there exists a partition since two new elements cannot be in the same partition. ▪

24



Randall Munro
http://xkcd.com/c287.html

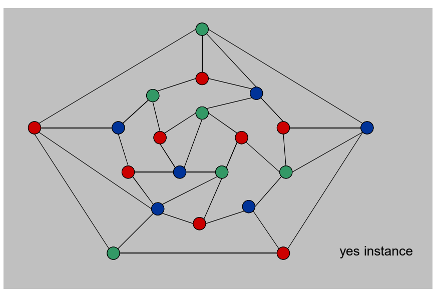
8.7 Graph Coloring

Basic genres.

- Packing problems: SET-PACKING, INDEPENDENT SET.
- Covering problems: SET-COVER, VERTEX-COVER.
- Constraint satisfaction problems: SAT, 3-SAT.
- Sequencing problems: HAMILTONIAN-CYCLE, TSP.
- Partitioning problems: 3D-MATCHING, 3-COLOR.
- Numerical problems: SUBSET-SUM, KNAPSACK.

3-Colorability

3-COLOR: Given an undirected graph G does there exists a way to color the nodes red, green, and blue so that no adjacent nodes have the same color?



Register Allocation

Register allocation. Assign program variables to machine register so that no more than k registers are used and no two program variables that are needed at the same time are assigned to the same register.

Interference graph. Nodes are program variables names, edge between u and v if there exists an operation where both u and v are "live" at the same time.

Observation. [Chaitin 1982] Can solve register allocation problem iff interference graph is k-colorable.

Fact. 3-COLOR $\le_P$ k-REGISTER-ALLOCATION for any constant $k \ge 3$.

3-Colorability

Claim. 3-SAT $\le_P$ 3-COLOR.

Pf. Given 3-SAT instance Φ, we construct an instance of 3-COLOR that is 3-colorable iff Φ is satisfiable.

Construction.

i. For each literal, create a node.
ii. Create 3 new nodes T, F, B; connect them in a triangle, and connect each literal to B.
iii. Connect each literal to its negation.
iv. For each clause, add gadget of 6 nodes and 13 edges.

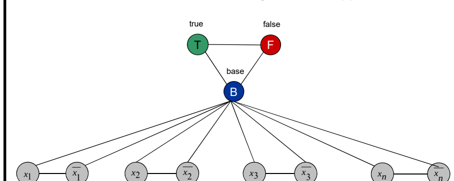
↑ to be described next

3-Colorability

Claim. Graph is 3-colorable iff Φ is satisfiable.

Pf. $\Rightarrow$ Suppose graph is 3-colorable.

- Consider assignment that sets all T literals to true.
- (ii) ensures each literal is T or F.
- (iii) ensures a literal and its negation are opposites.

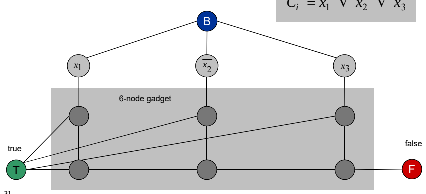


3-Colorability

Claim. Graph is 3-colorable iff Φ is satisfiable.

Pf. $\Rightarrow$ Suppose graph is 3-colorable.
- Consider assignment that sets all T literals to true.
- (ii) ensures each literal is T or F.
- (iii) ensures a literal and its negation are opposites.
- (iv) ensures at least one literal in each clause is T.

$$C_i = x_1 \vee \overline{x_2} \vee x_3$$

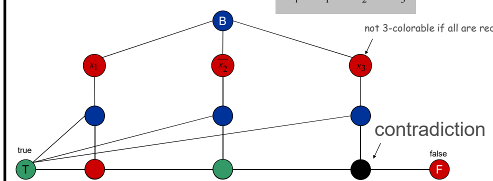


3-Colorability

Claim. Graph is 3-colorable iff Φ is satisfiable.

Pf. $\Rightarrow$ Suppose graph is 3-colorable.
- Consider assignment that sets all T literals to true.
- (ii) ensures each literal is T or F.
- (iii) ensures a literal and its negation are opposites.
- (iv) ensures at least one literal in each clause is T.

$$C_i = x_1 \vee \overline{x_2} \vee x_3$$

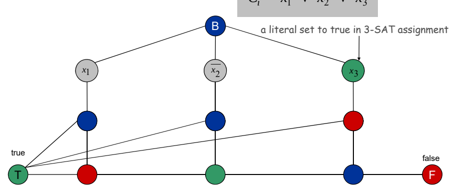


3-Colorability

Claim. Graph is 3-colorable iff Φ is satisfiable.

Pf. $\Leftarrow$ Suppose 3-SAT formula Φ is satisfiable.
- Color all true literals T.
- Color node below green node F, and node below that B.
- Color remaining middle row nodes B.
- Color remaining bottom nodes T or F as forced. ▪

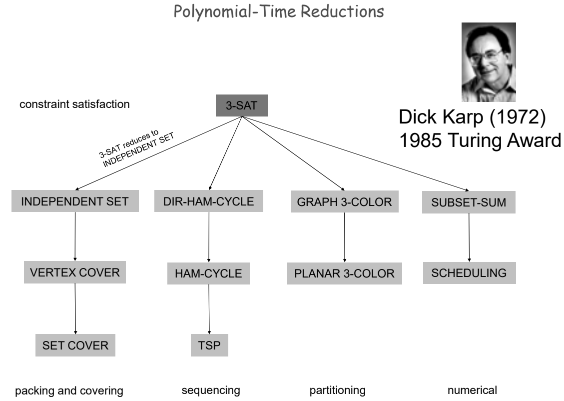
$$C_i = x_1 \vee \overline{x_2} \vee x_3$$



Extra Slides

8.10 A Partial Taxonomy of Hard Problems

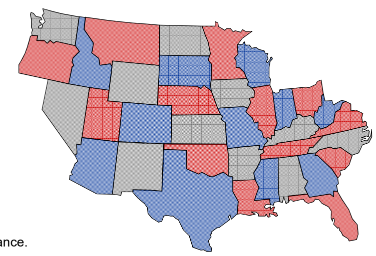
Polynomial-Time Reductions



4 Color Theorem

Planar 3-Colorability

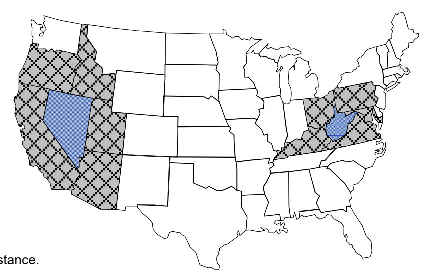
PLANAR-3-COLOR. Given a planar map, can it be colored using 3 colors so that no adjacent regions have the same color?



YES instance.

Planar 3-Colorability

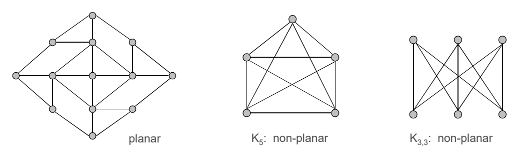
PLANAR-3-COLOR. Given a planar map, can it be colored using 3 colors so that no adjacent regions have the same color?



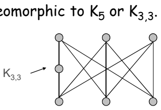
NO instance.

Planarity

Def. A graph is planar if it can be embedded in the plane in such a way that no two edges cross.
Applications: VLSI circuit design, computer graphics.



Kuratowski's Theorem. An undirected graph G is non-planar iff it contains a subgraph homeomorphic to K_5 or $K_{3,3}$.



Planarity Testing

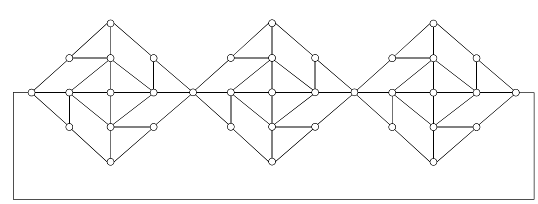
Planarity testing. [Hopcroft-Tarjan 1974] O(n).

simple planar graph can have at mo

Remark. Many intractable graph problems can be solved in poly-time if the graph is planar; many tractable graph problems can be solved faster if the graph is planar.

Planar Graph 3-Colorability

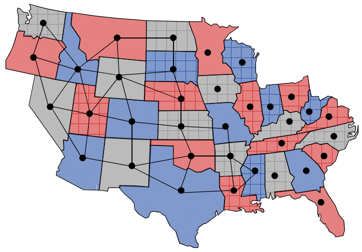
Q. Is this planar graph 3-colorable?



Planar 3-Colorability and Graph 3-Colorability

Claim. PLANAR-3-COLOR $\leq_P$ PLANAR-GRAPH-3-COLOR.

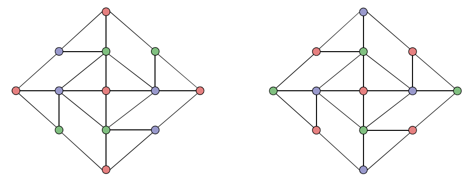
Pf sketch. Create a vertex for each region, and an edge between regions that share a nontrivial border.



Planar Graph 3-Colorability

Claim. W is a planar graph such that:
- In any 3-coloring of W, opposite corners have the same color.
- Any assignment of colors to the corners in which opposite corners have the same color extends to a 3-coloring of W.

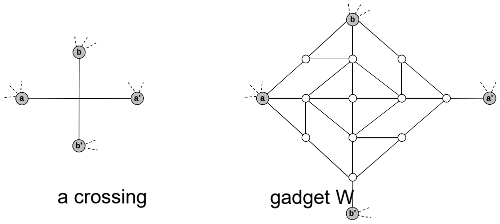
Pf. Only 3-colorings of W are shown below (or by permuting colors).



Planar Graph 3-Colorability

Claim. 3-COLOR $\leq_P$ PLANAR-GRAPH-3-COLOR.

Pf. Given instance of 3-COLOR, draw graph in plane, letting edges cross.
- Replace each edge crossing with planar gadget W.
- In any 3-coloring of W, a $\neq$ a' and b $\neq$ b'.
- If a $\neq$ a' and b $\neq$ b' then can extend to a 3-coloring of W.



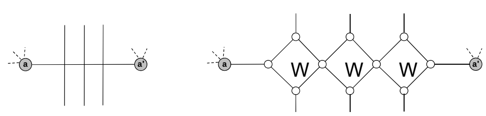
a crossing

gadget W

Planar Graph 3-Colorability

Claim. 3-COLOR $\leq_P$ PLANAR-GRAPH-3-COLOR.

Pf. Given instance of 3-COLOR, draw graph in plane, letting edges cross.
- Replace each edge crossing with planar gadget W.
- In any 3-coloring of W, a $\neq$ a' and b $\neq$ b'.
- If a $\neq$ a' and b $\neq$ b' then can extend to a 3-coloring of W.



multiple crossings

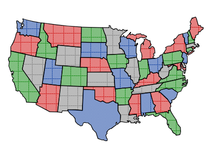
gadget W

Planar k-Colorability

PLANAR-2-COLOR. Solvable in linear time.

PLANAR-3-COLOR. NP-complete.

PLANAR-4-COLOR. Solvable in O(1) time.



Theorem. [Appel-Haken, 1976] Every planar map is 4-colorable.
- Resolved century-old open problem.
- Used 50 days of computer time to deal with many special cases.
- First major theorem to be proved using computer.

False intuition. If PLANAR-3-COLOR is hard, then so is PLANAR-4-COLOR and PLANAR-5-COLOR.

Polynomial-Time Detour

Graph minor theorem. [Robertson-Seymour 1980s]

Corollary. There exist an $O(n^3)$ algorithm to determine if a graph can be embedded in the torus in such a way that no two edges cross.

Pf of theorem. Tour de force.

Polynomial-Time Detour

Graph minor theorem. [Robertson-Seymour 1980s]

Corollary. There exist an O(n^3) algorithm to determine if a graph can be embedded in the torus in such a way that no two edges cross.

Mind boggling fact 1. The proof is highly non-constructive!
Mind boggling fact 2. The constant of proportionality is enormous!

> Unfortunately, for any instance G = (V, E) that one could fit into the known universe, one would easily prefer n^{70} to even *constant* time, if that constant had to be one of Robertson and Seymour's. - David Johnson

Theorem. There exists an explicit O(n) algorithm.
Practice. LEDA implementation guarantees O(n^3).