Data Structures and Algorithms

Session 21. April 13, 2009

Instructor: Bert Huang

http://www.cs.columbia.edu/~bert/courses/3137

Announcements

- Homework 5 due next Monday
- I’m out of town Wed to Sun for conference
- Course topic order changed.
  - After today: Disjoint Sets, Sorting

Review

- Minimum Spanning Tree
  - Prim's Algorithm
  - Kruskal's Algorithm
- Depth first search
  - Euler Paths
- Hamiltonian Path/Cycle

Today's Plan

- High Level Introduction to Complexity Classes
  - P, NP, NP-Complete, NP-hard, Undecidable
  - Famous NP-Complete problems

Hamiltonian Cycle

- Now that we know how to find Euler circuits efficiently, can we find Hamiltonian Cycles?
- Hamiltonian cycle - path that visits each *node* once, starts and ends on same node
- Can we return true/false if a Hamiltonian Cycle exists?

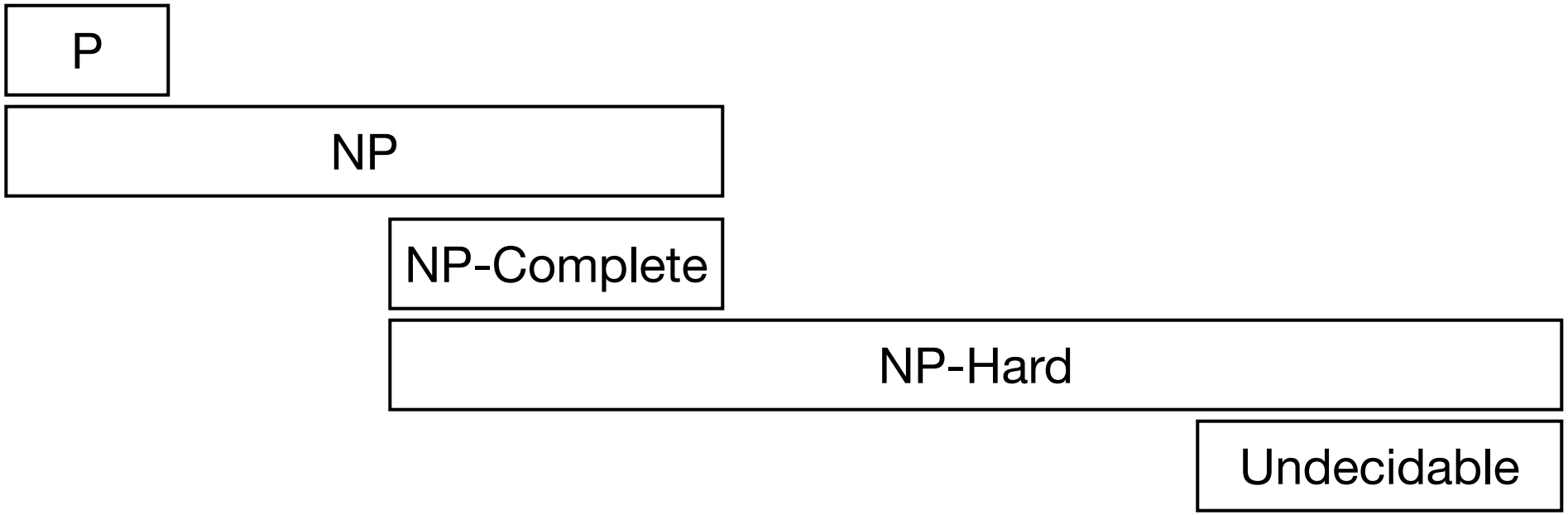
Hamiltonian Cycle

- One easy necessary condition: every node must have at least 2 edges. Anything more specific?
- We can't splice cycles together like Euler Circuits, since we can't revisit a node
- We can't use the edge-graph because edges connect to multiple nodes, but not vice versa

Complexity Classes

- **P** - solvable in polynomial time
- **NP** - solvable in polynomial time by a nondeterministic computer
  - i.e., you can check a solution in polynomial time
- **NP-complete** - a problem in NP such that any problem in NP is polynomially reducible to it
- **Undecidable** - no algorithm can solve the problem

Probable Complexity Class Hierarchy



Polynomial Time **P**

- All the algorithms we cover in class are solvable in polynomial time
- An algorithm that runs in polynomial time is considered **efficient**
- A problem solvable in polynomial time is considered **tractable**

Nondeterministic Polynomial Time **NP**

- Consider a magical nondeterministic computer
  - infinitely parallel computer
- Equivalently, to solve any problem, check every possible solution in parallel
  - return one that passes the check

NP-Complete

- Special class of NP problems that can be used to solve any other NP problem
- Hamiltonian Path, Satisfiability, Graph Coloring etc.
- NP-Complete problems can be **reduced** to other NP-Complete problems:
  - polynomial time algorithm to convert the input and output of algorithms

NP-Hard

- A problem is NP-Hard if it is at least as complex as all NP-Complete problems
- NP-hard problems may not even be NP

Undecidable

- No algorithm can solve undecidable problems
- **halting problem -** given a computer program, determine if the computer program will finish
- Sketch of undecidability proof: Assume we have an algorithm that detects infinite loops
  - Write program LOOP(P) that runs P on itself
  - If P(P) halts, infinite loop, otherwise output

Halting Problem

```
LOOP(P):
   If P(P) will halt:        infinite loop
   If P(P) runs forever: output
```

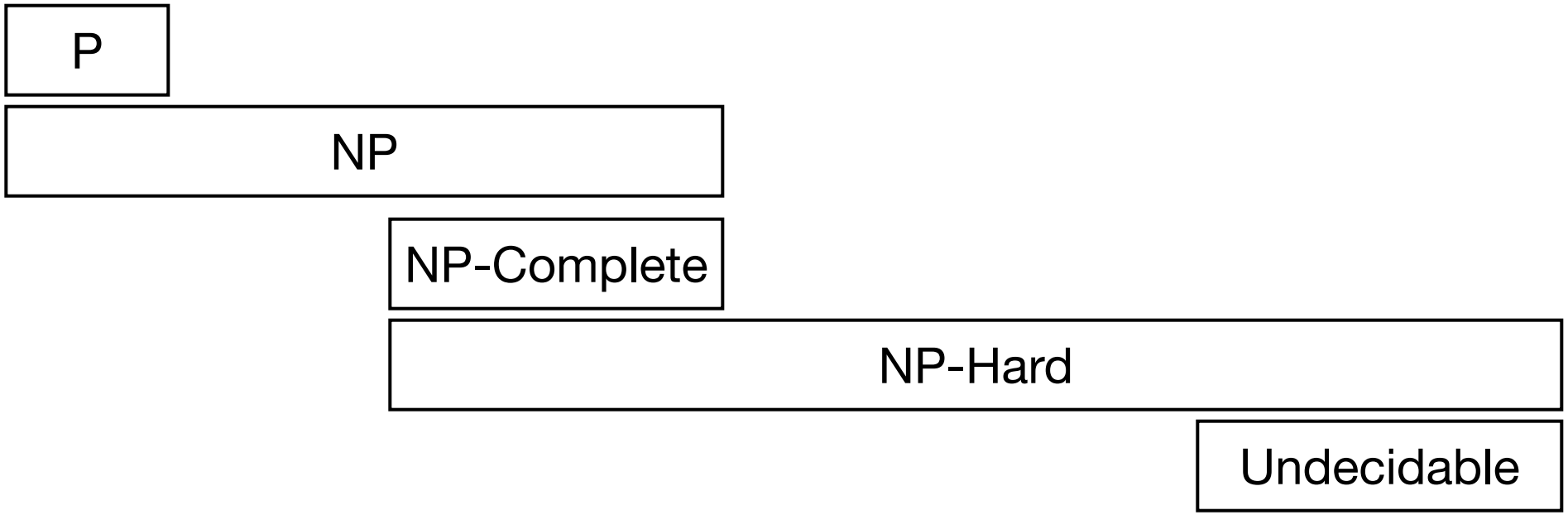
- What happens if we call LOOP(LOOP)?

```
LOOP(LOOP):
   If LOOP(LOOP) will halt:        infinite loop
   If LOOP(LOOP) runs forever: output
```

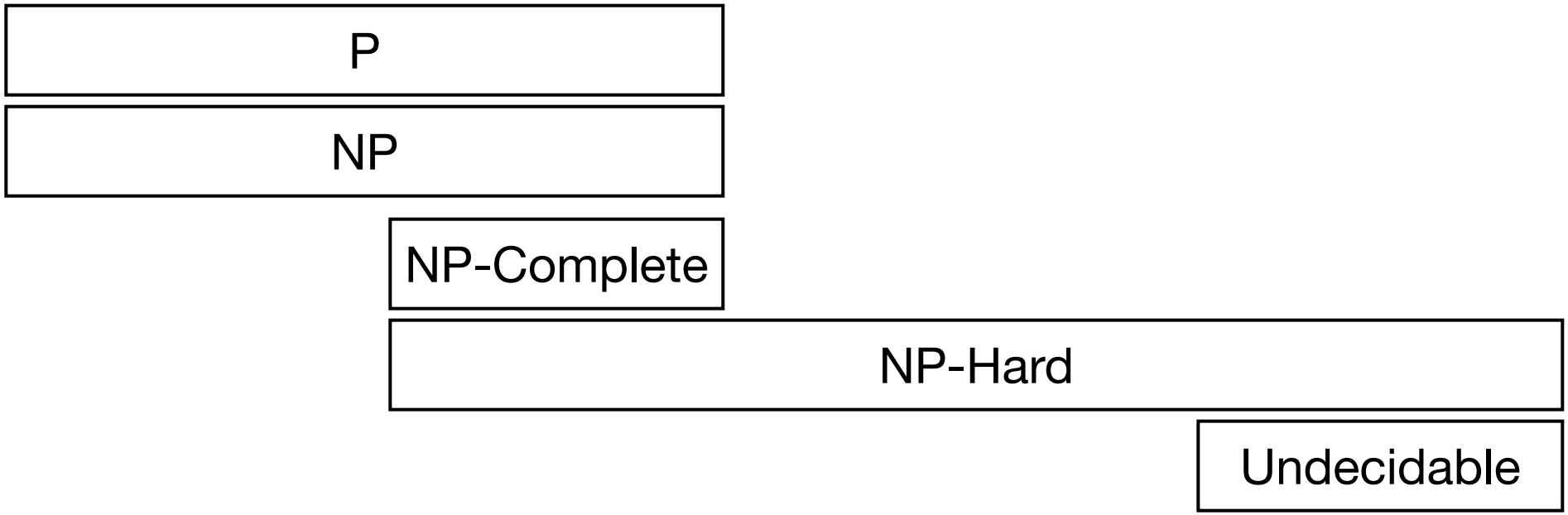
P versus NP

- So far, nobody has proven a super-polynomial lower bound on any NP-Complete problems,
- nor has anyone found a polynomial time algorithm for an NP-complete problem
- Therefore no problem is proven to be in one class but not the other;
  - it is unknown if P and NP are the same

Probable Complexity Class Hierarchy



Unlikely Complexity Class Hierarchy



- ...but theoretically still possible as far as we know

NP-Complete Problems Satisfiability

- Given Boolean expression of N variables, can we set variables to make expression true?
- First NP-Complete proof because Cook's Theorem gave polynomial time procedure to convert any NP problem to a Boolean expression
- I.e., if we have efficient algorithm for Satisfiability, we can efficiently solve any NP problem

NP-Complete Problems Graph Coloring

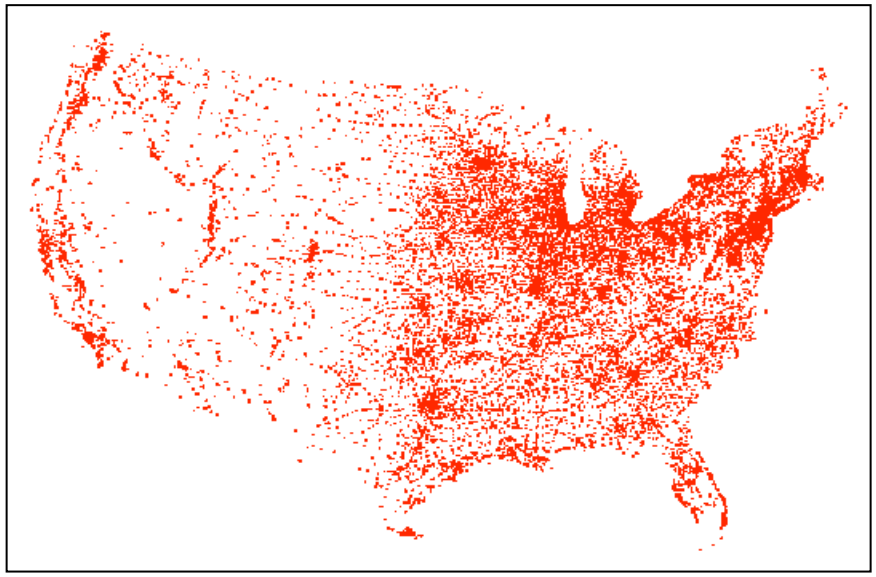
- Given a graph is it possible to color with **k** colors all nodes so no adjacent nodes are the same color?
- Coloring countries on a map
- Sudoku is a form of this problem. All squares in a row, column and blocks are connected. **k** = 9

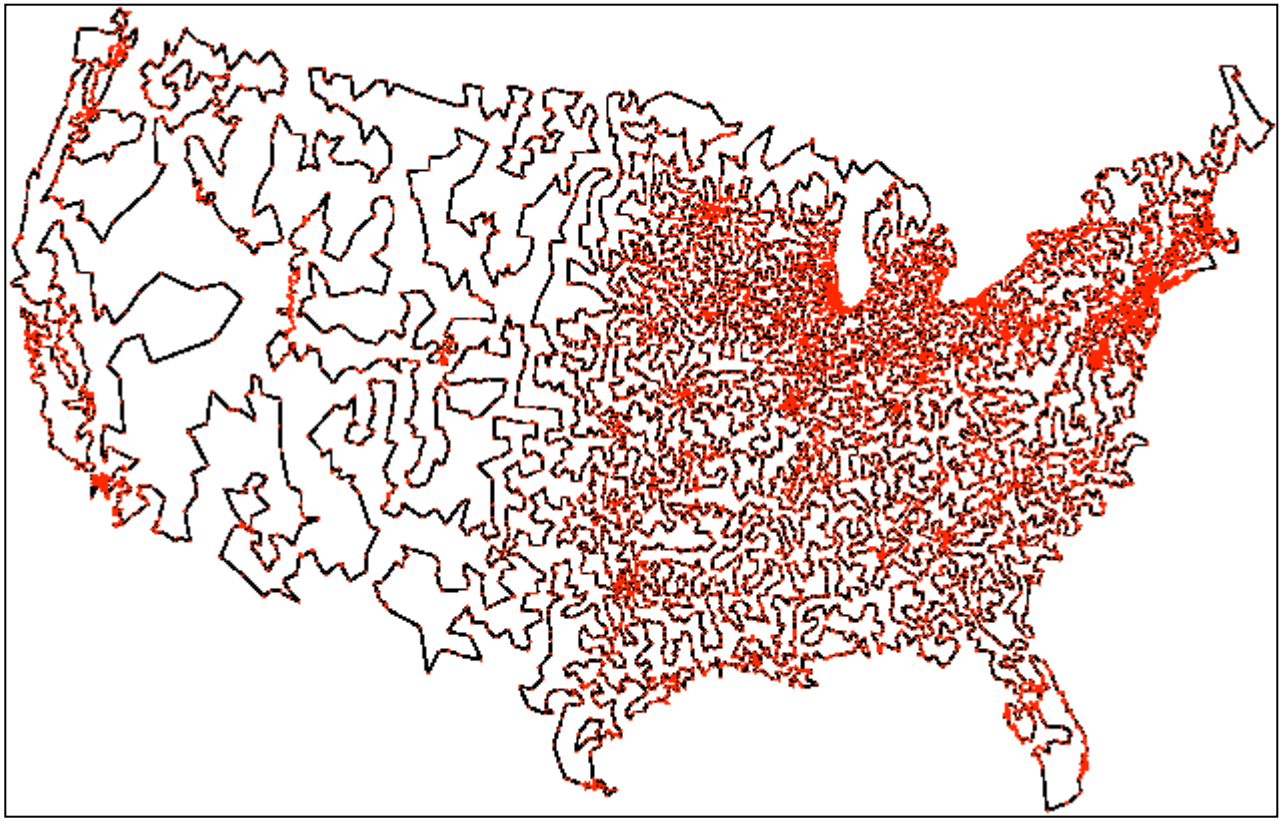
NP-Complete Problems
Hamiltonian Path

- Given a graph with N nodes, is there a path that visits each node exactly once?

NP-Hard Problems Traveling Salesman

- Closely related to Hamiltonian Path problem
- Given complete graph **G**, find a path that visits all nodes that costs less than some constant **k**
- If we are able to solve TSP, we can find a Hamiltonian Path; set connected edge weight to constant, disconnected to infinity
  - TSP is NP-hard





http://www.tsp.gatech.edu/gallery/itours/usa13509.html

Reading

- Weiss 9.7
- http://www.tsp.gatech.edu/
- Weiss Chapter 8
  Disjoint Set ADT (Next class)