

CSE 417

Algorithms and Complexity

Autumn 2020

Lecture 30

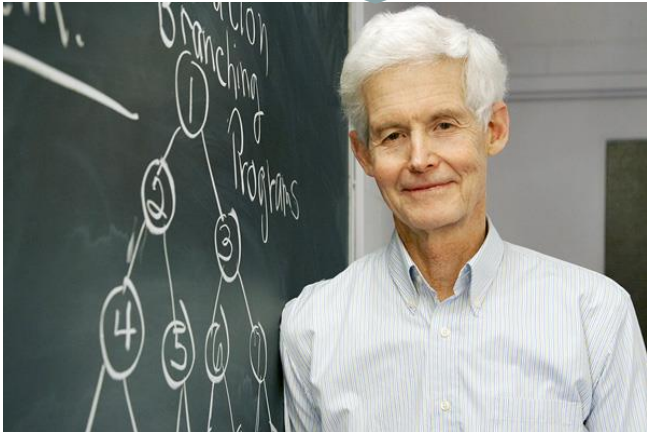
NP-Completeness and Beyond

Announcements

- Final Exam: Monday, December 14
 - 24 hour take home exam
 - Target: 2 to 4 hours of work time
 - Available at 2:30 PM, Monday, Dec 14
 - Due 2:29 PM, Tuesday, Dec 15
 - Open book / notes.
- Extra office hours
- Approximate grade weighting
 - 75% HW, 25% Final

NP Completeness: The story so far

Circuit Satisfiability is
NP-Complete

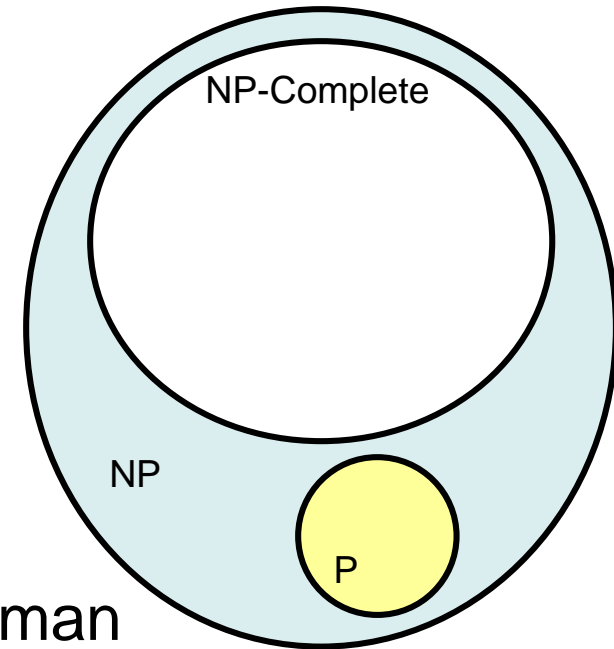


There are a whole bunch of
other important problems
which are NP-Complete



Populating the NP-Completeness Universe

- Circuit Sat \leq_p 3-SAT
- 3-SAT \leq_p Independent Set
- 3-SAT \leq_p Vertex Cover
- Independent Set \leq_p Clique
- 3-SAT \leq_p Hamiltonian Circuit
- Hamiltonian Circuit \leq_p Traveling Salesman
- 3-SAT \leq_p Integer Linear Programming
- 3-SAT \leq_p Graph Coloring
- 3-SAT \leq_p Subset Sum
- Subset Sum \leq_p Scheduling with Release times and deadlines



Satisfiability

Literal: A Boolean variable or its negation.

$$x_i \text{ or } \overline{x_i}$$

Clause: A disjunction of literals.

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

Conjunctive normal form: A propositional formula Φ that is the conjunction of clauses.

$$\Phi = C_1 \wedge C_2 \wedge C_3 \wedge C_4$$

SAT: Given CNF formula Φ , does it have a satisfying truth assignment?

3-SAT: SAT where each clause contains exactly 3 literals.

Ex: $(\overline{x_1} \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$

Yes: $x_1 = \text{true}, x_2 = \text{true}, x_3 = \text{false}.$

Today

REDUCIBILITY AMONG COMBINATORIAL PROBLEMS[†]

Richard M. Karp

University of California at Berkeley

There are a whole bunch of other important problems which are NP-Complete



Abstract: A large class of computational problems involve the determination of properties of graphs, digraphs, integers, arrays of integers, finite families of finite sets, boolean formulas and elements of other countable domains. Through simple encodings from such domains into the set of words over a finite alphabet these problems can be converted into language recognition problems, and we can inquire into their computational complexity. It is reasonable to consider such a problem satisfactorily solved when an algorithm for its solution is found which terminates within a number of steps bounded by a polynomial in the length of the input. We show that a large number of classic unsolved problems of covering, matching, packing, routing, assignment and sequencing are equivalent, in the sense that either each of them possesses a polynomial-bounded algorithm or none of them does.

1. INTRODUCTION

All the general methods presently known for computing the chromatic number of a graph, deciding whether a graph has a Hamilton circuit, or solving a system of linear inequalities in which the variables are constrained to be 0 or 1, require a combinatorial search for which the worst case time requirement grows exponentially with the length of the input. In this paper we give theorems which strongly suggest, but do not imply, that these problems, as well as many others, will remain intractable perpetually.

[†]This research was partially supported by National Science Foundation Grant GJ-474.

Reducibility Among Combinatorial Problems

96

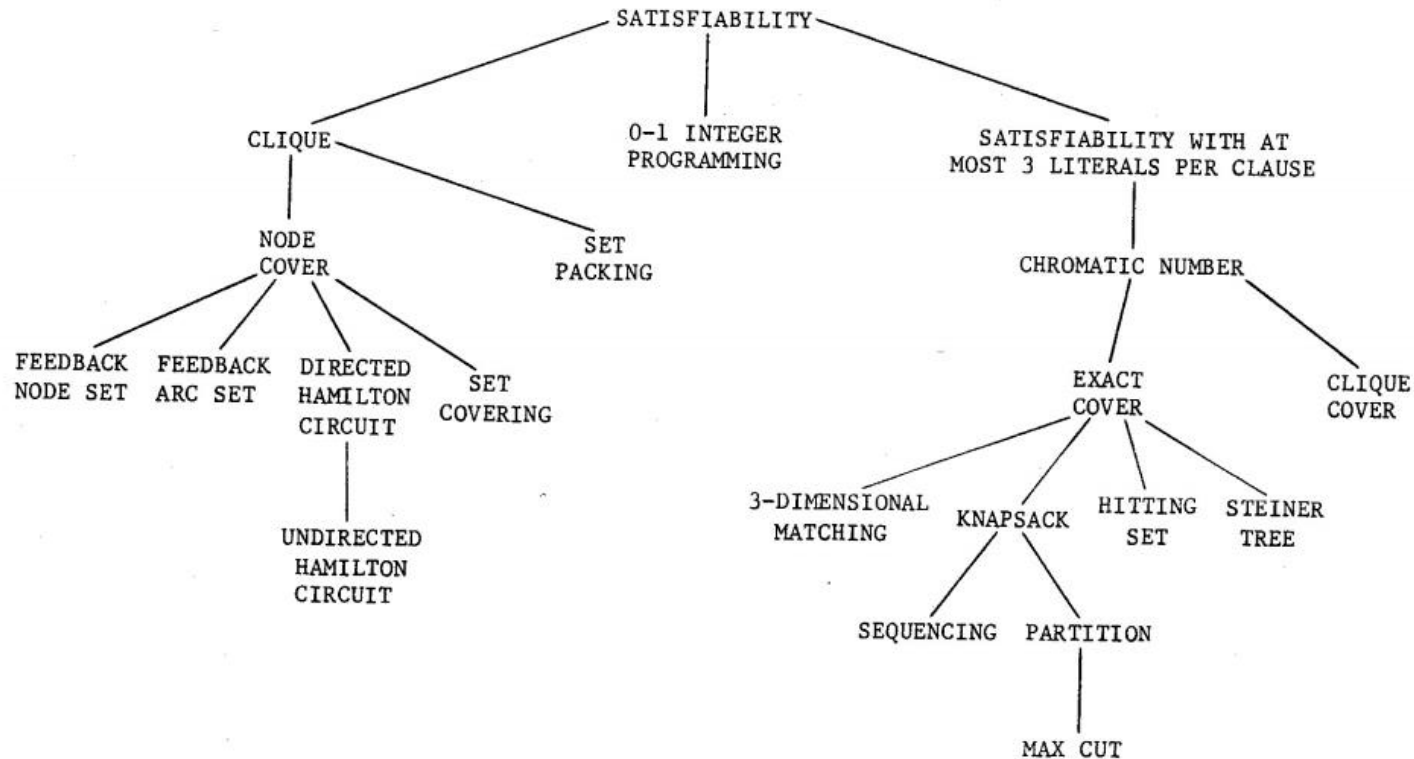
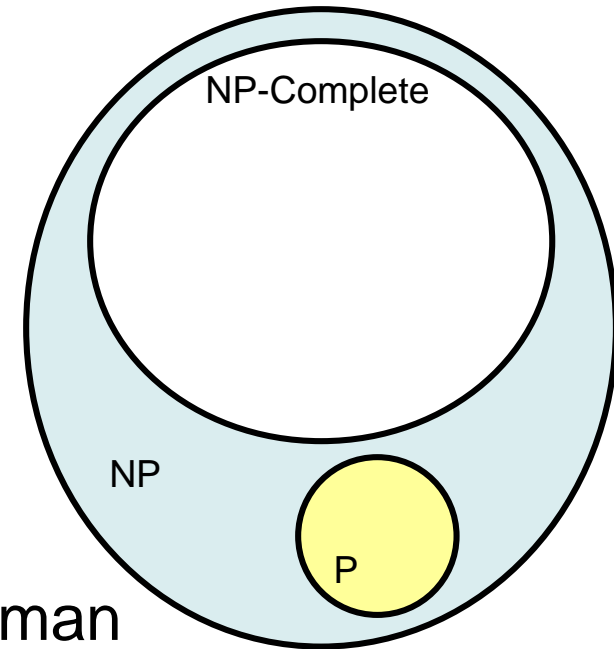


FIGURE 1 - Complete Problems

Populating the NP-Completeness Universe

- Circuit Sat \leq_p 3-SAT
- 3-SAT \leq_p Independent Set
- 3-SAT \leq_p Vertex Cover
- Independent Set \leq_p Clique
- 3-SAT \leq_p Hamiltonian Circuit
- Hamiltonian Circuit \leq_p Traveling Salesman
- 3-SAT \leq_p Integer Linear Programming
- 3-SAT \leq_p Graph Coloring
- 3-SAT \leq_p Subset Sum
- Subset Sum \leq_p Scheduling with Release times and deadlines



NP-Completeness Proofs

- Prove that problem X is NP-Complete
 - Show that X is in NP (usually easy)
 - Pick a known NP complete problem Y
 - Show $Y <_p X$

Satisfiability

Literal: A Boolean variable or its negation.

$$x_i \text{ or } \overline{x_i}$$

Clause: A disjunction of literals.

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

Conjunctive normal form: A propositional formula Φ that is the conjunction of clauses.

$$\Phi = C_1 \wedge C_2 \wedge C_3 \wedge C_4$$

SAT: Given CNF formula Φ , does it have a satisfying truth assignment?

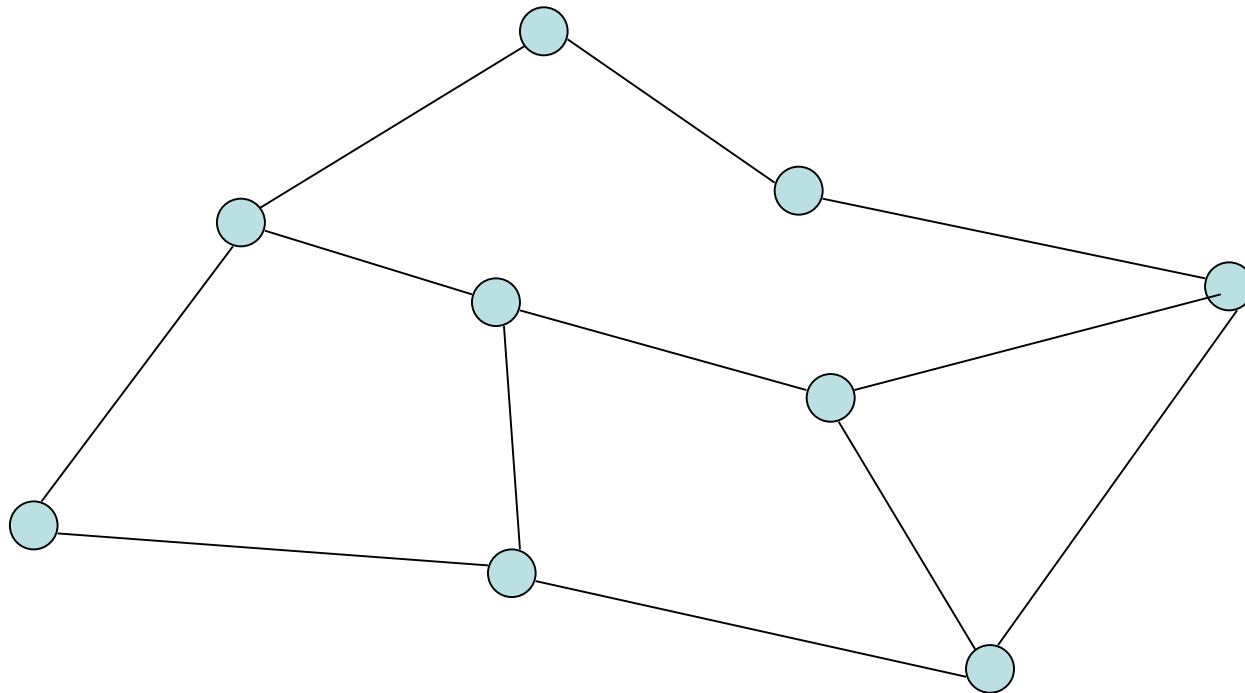
3-SAT: SAT where each clause contains exactly 3 literals.

Ex: $(\overline{x_1} \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$

Yes: $x_1 = \text{true}, x_2 = \text{true}, x_3 = \text{false}.$

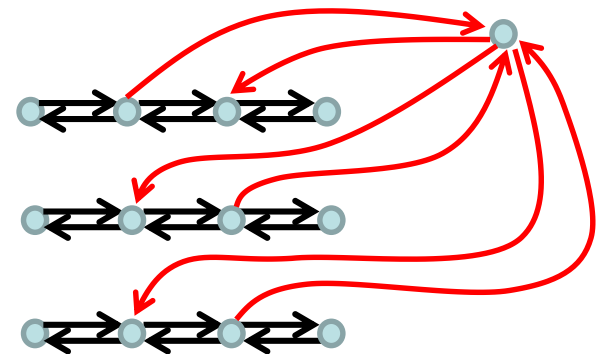
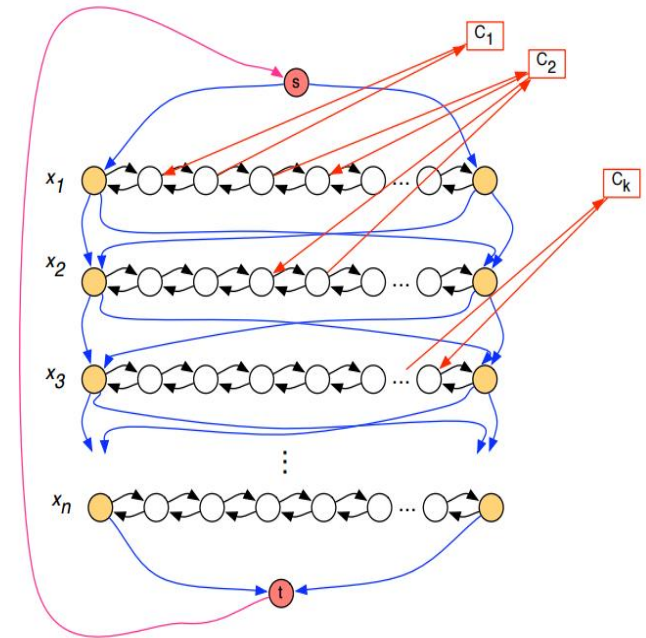
Hamiltonian Circuit Problem

- Hamiltonian Circuit – a simple cycle including all the vertices of the graph



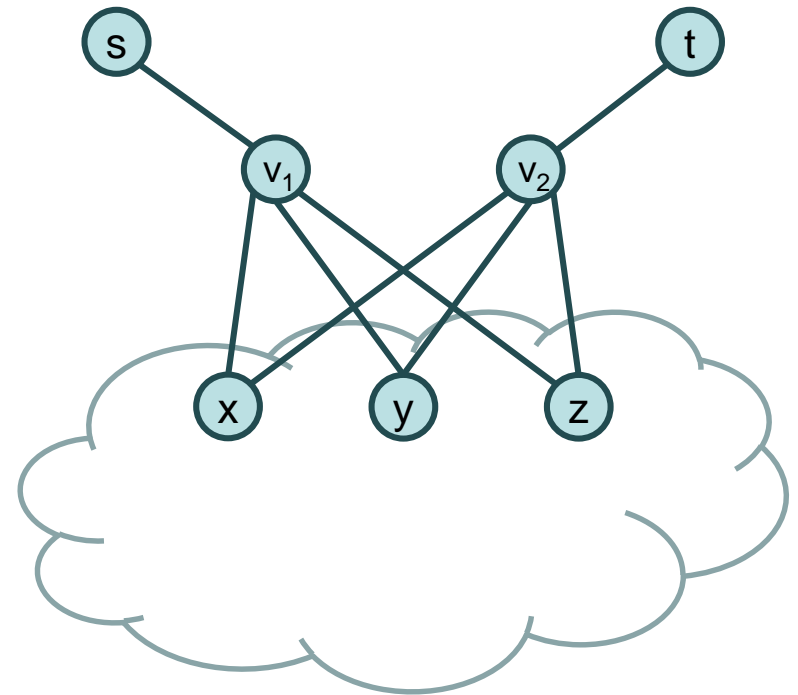
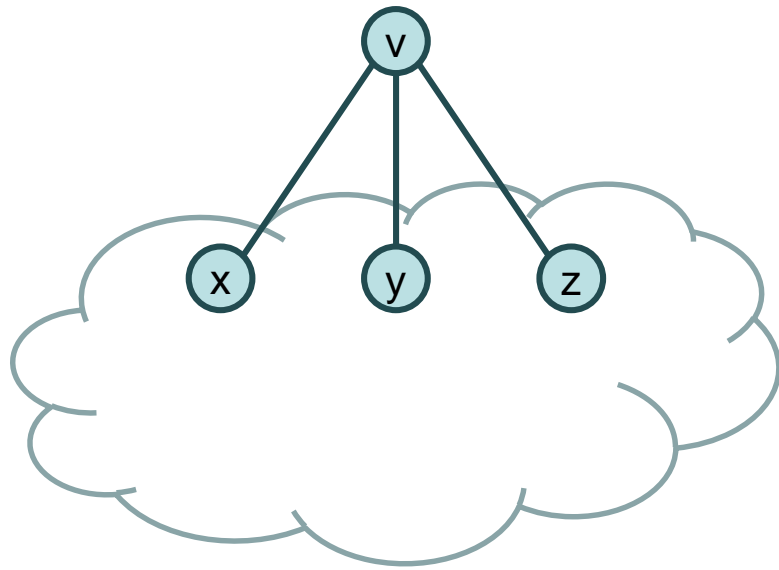
Thm: Hamiltonian Circuit is NP Complete

- Reduction from 3-SAT
- Formula F to Graph G
- G has a Hamiltonian Circuit IFF F has a satisfying assignment
- See Page 475 in Text



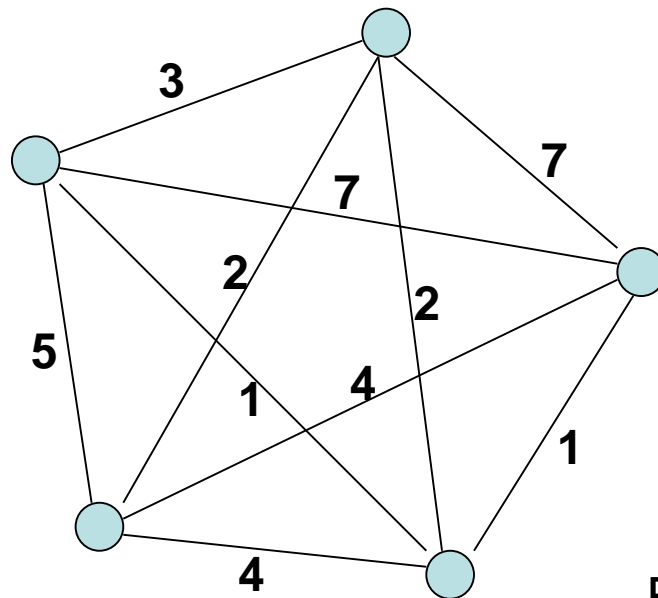
Reduce Hamiltonian Circuit to Hamiltonian Path

G_2 has a Hamiltonian Path iff G_1 has a Hamiltonian Circuit



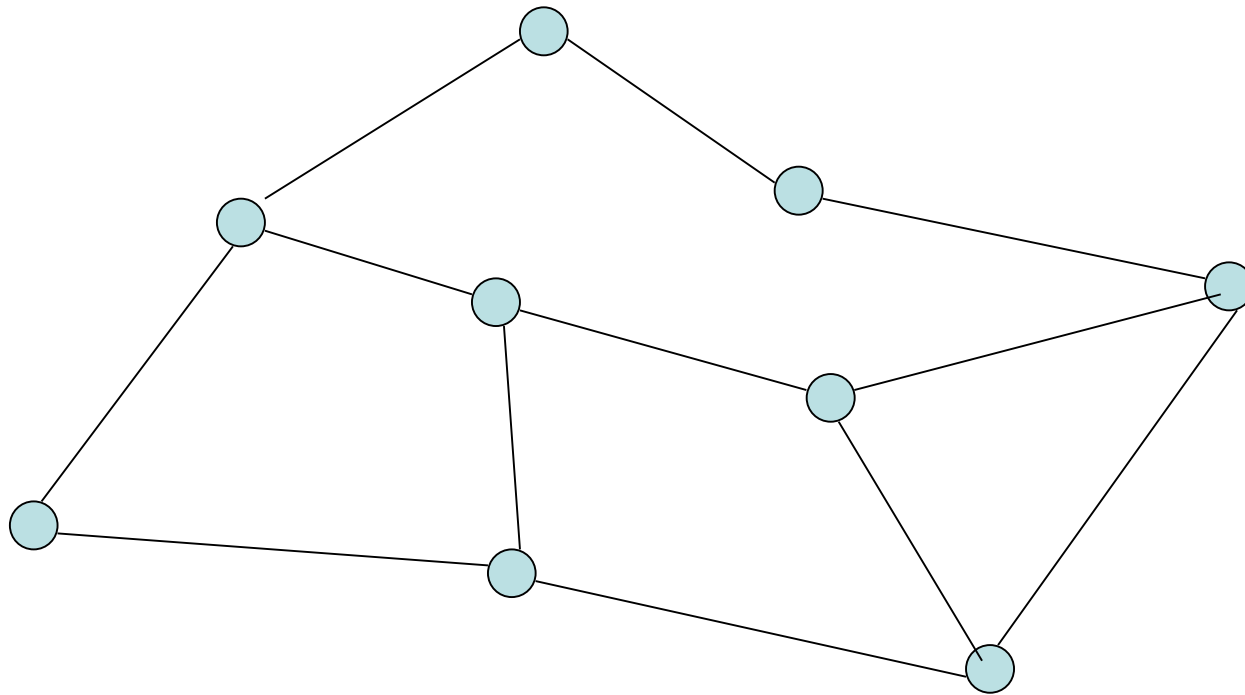
Traveling Salesman Problem

- Given a complete graph with edge weights, determine the shortest tour that includes all of the vertices (visit each vertex exactly once, and get back to the starting point)

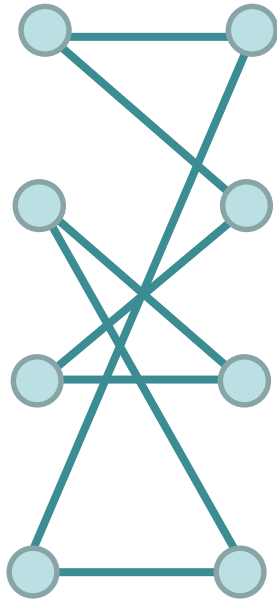


Find the minimum cost tour

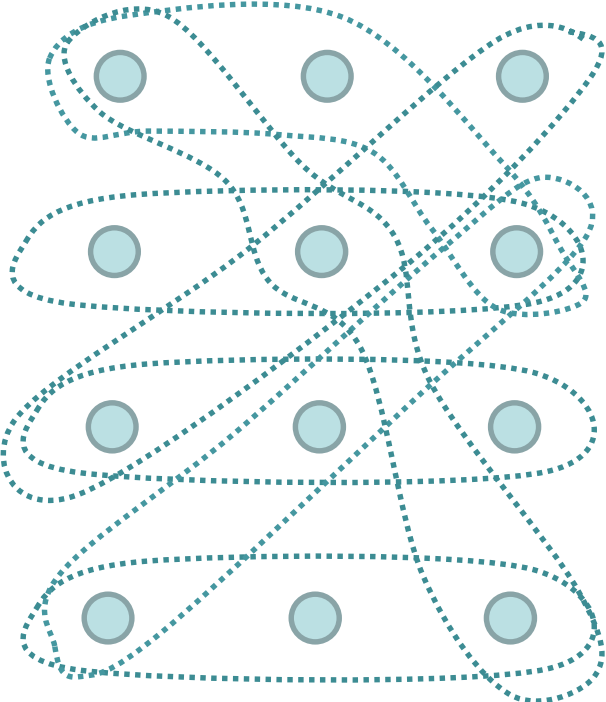
Thm: $HC \leq_p TSP$



Matching



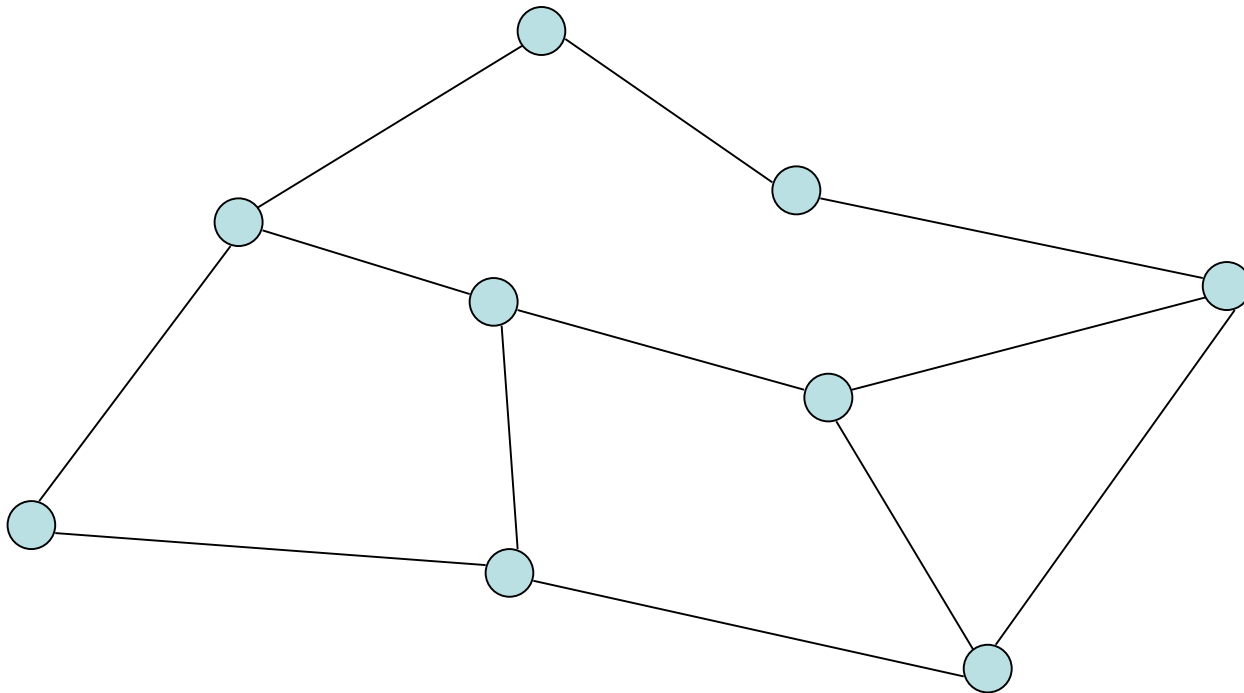
Two dimensional
matching



Three dimensional
matching (3DM)

Graph Coloring

- NP-Complete
 - Graph K-coloring
 - Graph 3-coloring
- Polynomial
 - Graph 2-Coloring



Number Problems

- Subset sum problem
 - Given natural numbers w_1, \dots, w_n and a target number W , is there a subset that adds up to exactly W ?
- Subset sum problem is NP-Complete
- Subset Sum problem can be solved in $O(nW)$ time

Integer Linear Programming

- Linear Programming – maximize a linear function subject to linear constraints
- Integer Linear Programming – require an integer solution
- NP Completeness reduction from 3-SAT

Use 0-1 variables for x_i 's

Constraint for clause $x_1 \vee \overline{x_2} \vee \overline{x_3}$

$$x_1 + (1 - x_2) + (1 - x_3) > 0$$

Coping with NP-Completeness

- Approximation Algorithms
- Exact solution via Branch and Bound
- Local Search

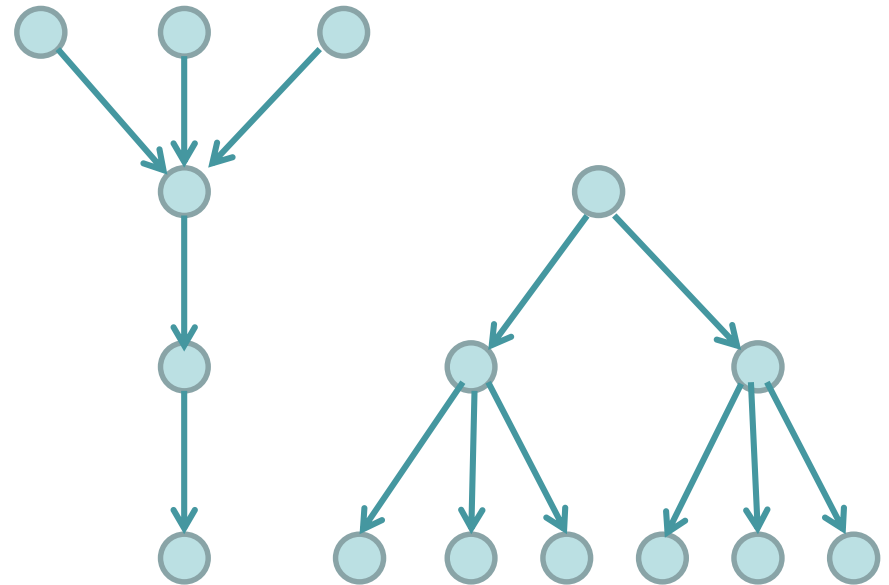


I can't find an efficient algorithm, but neither can all these famous people.

Multiprocessor Scheduling

- Unit execution tasks
- Precedence graph
- K-Processors

- Polynomial time for $k=2$
- Open for $k = \text{constant}$
- NP-complete if k is part of the problem

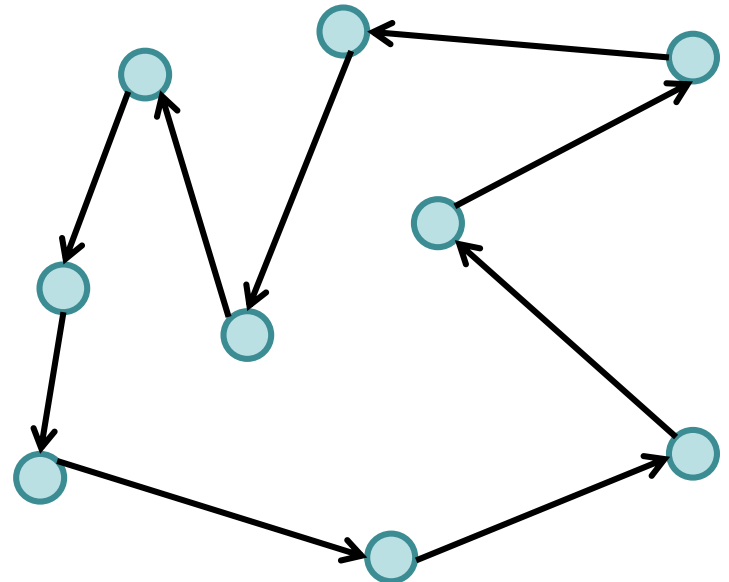
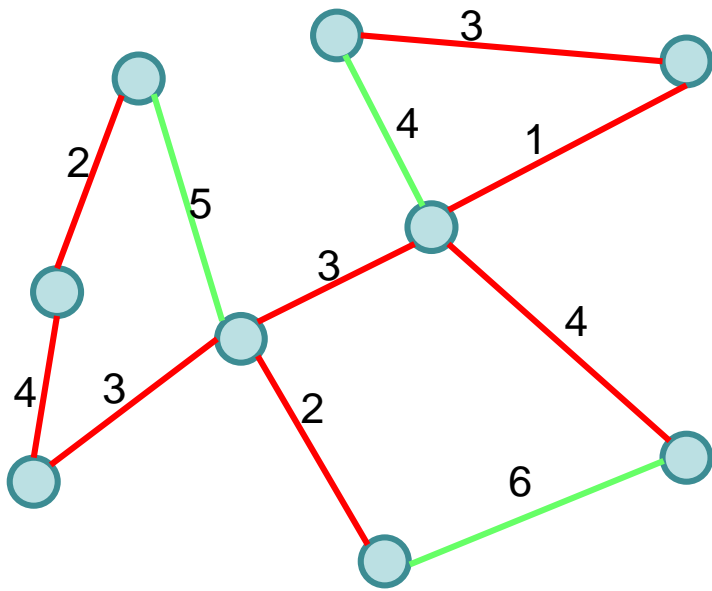
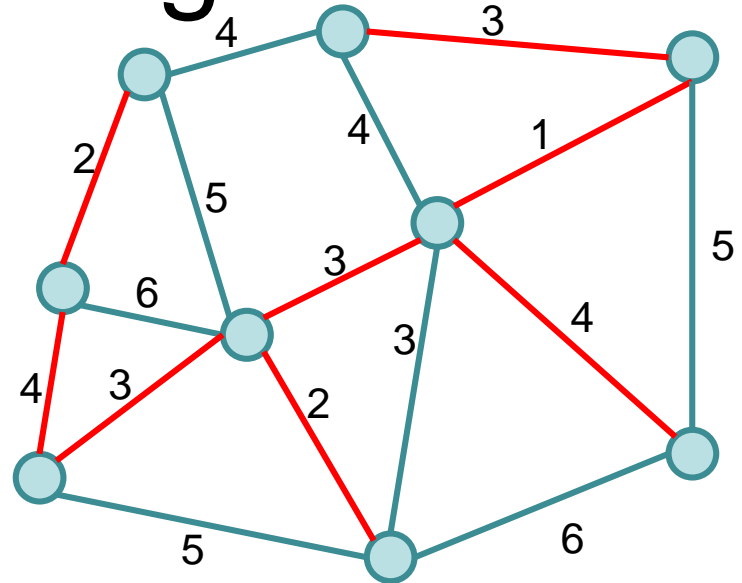
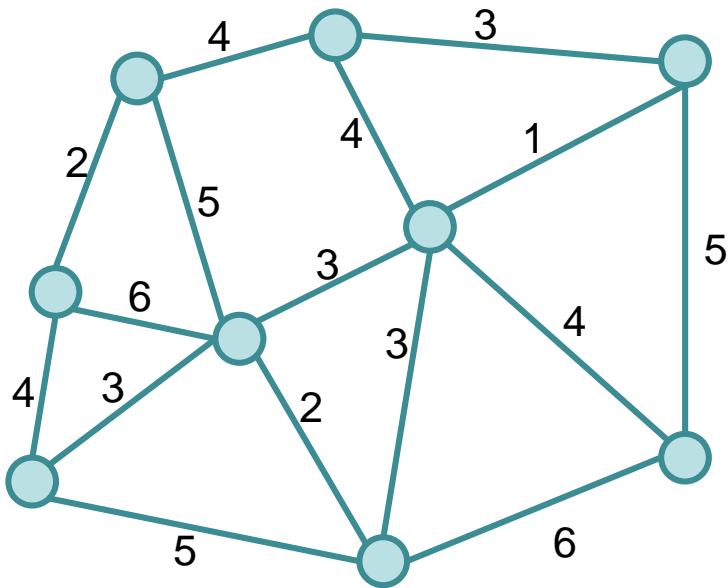


Highest level first is 2-Optimal

Choose k items on the highest level

Claim: number of rounds is at least twice the optimal.

Christofides Algorithm



Bin Packing

- Given N items with weight w_i , pack the items into as few unit capacity bins as possible
- Example: .3, .3, .3, .3, .4, .4

First Fit Packing

- First Fit
 - Theorem: $FF(I)$ is at most $17/10 \text{ Opt}(I) + 2$
- First Fit Decreasing
 - Theorem: $FFD(I)$ is at most $11/9 \text{ Opt}(I) + 4$

Branch and Bound

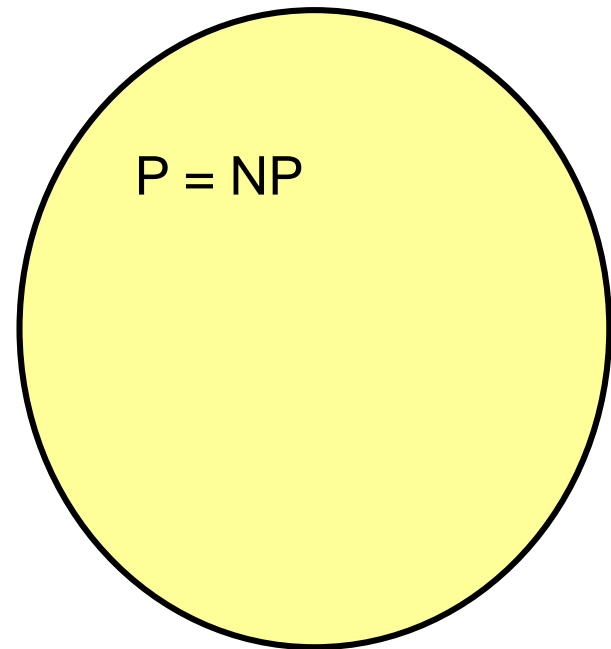
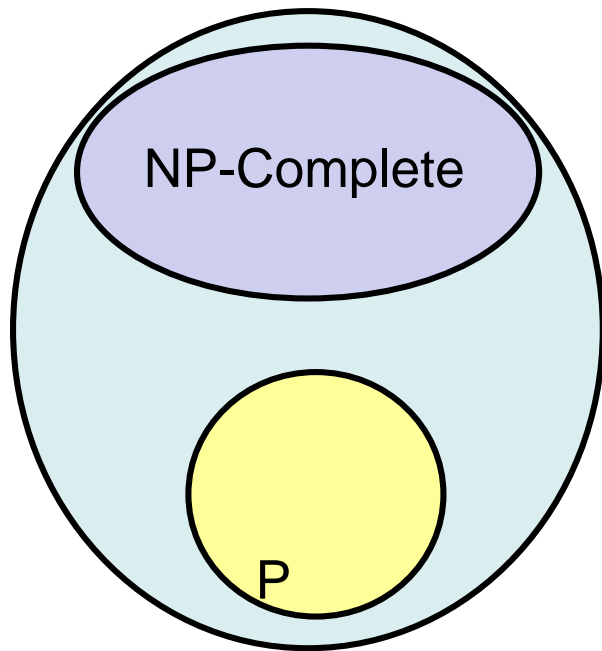
- Brute force search – tree of all possible solutions
- Branch and bound – compute a lower bound on all possible extensions
 - Prune sub-trees that cannot be better than optimal

Local Optimization

- Improve an optimization problem by local improvement
 - Neighborhood structure on solutions
 - Travelling Salesman 2-Opt (or K-Opt)
 - Independent Set Local Replacement

What we don't know

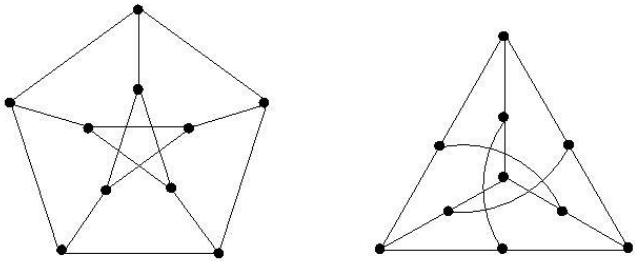
- P vs. NP



If $P \neq NP$, is there anything in between

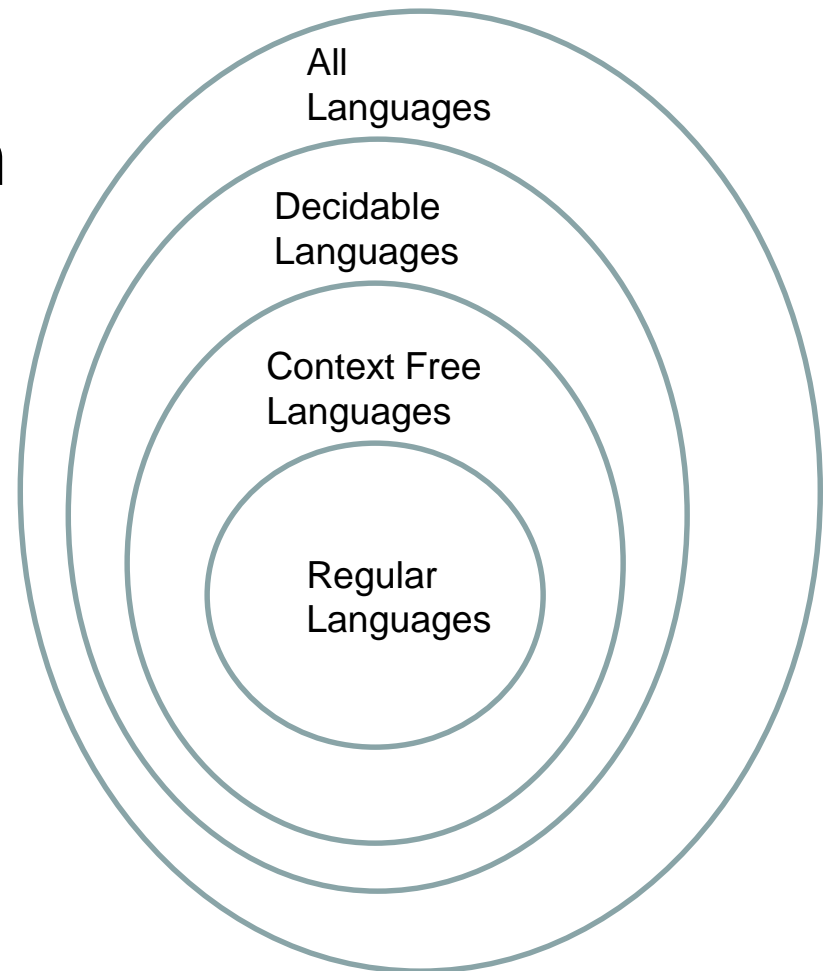
- Yes, Ladner [1975]
- Problems not known to be in P or NP Complete
 - Factorization
 - Discrete Log
 - Graph Isomorphism

Solve $g^k = b$ over a finite group



Complexity Theory

- Computational requirements to recognize languages
- Models of Computation
- Resources
- Hierarchies



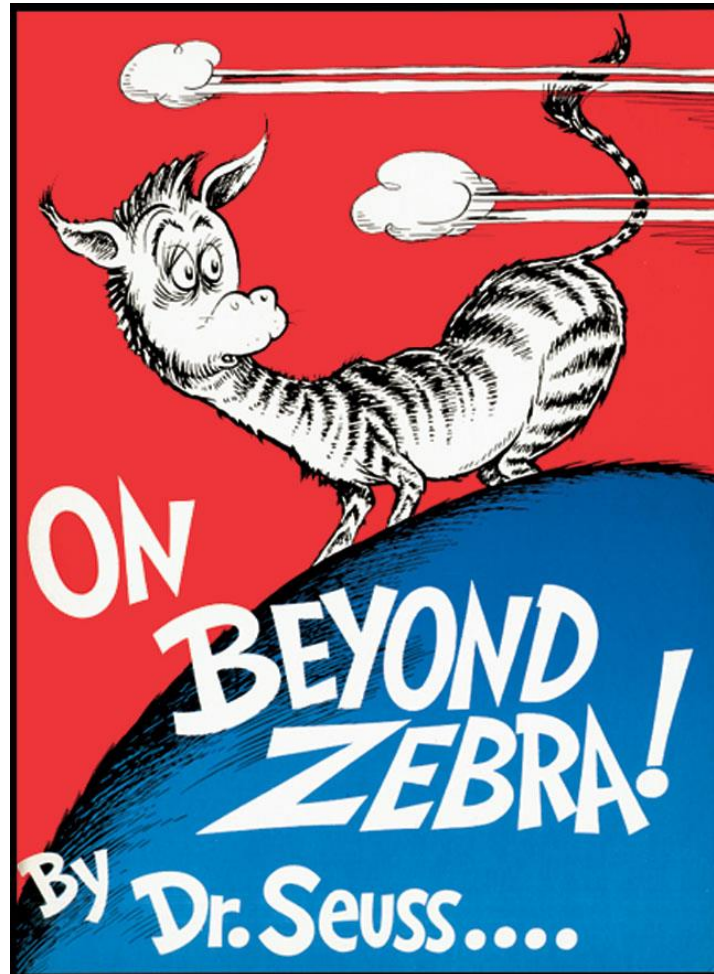
Time complexity

- P: (Deterministic) Polynomial Time
- NP: Non-deterministic Polynomial Time
- EXP: Exponential Time

Space Complexity

- Amount of Space (Exclusive of Input)
- L: Logspace, problems that can be solved in $O(\log n)$ space for input of size n
 - Related to Parallel Complexity
- PSPACE, problems that can be required in a polynomial amount of space

So what is beyond NP?



NP vs. Co-NP

- Given a Boolean formula, is it true for some choice of inputs
- Given a Boolean formula, is it true for all choices of inputs

Problems beyond NP

- Exact TSP, Given a graph with edge lengths and an integer K , does the minimum tour have length K
- Minimum circuit, Given a circuit C , is it true that there is no smaller circuit that computes the same function as C

Polynomial Hierarchy

- Level 1

- $\exists X_1 \Phi(X_1), \forall X_1 \Phi(X_1)$

- Level 2

- $\forall X_1 \exists X_2 \Phi(X_1, X_2), \exists X_1 \forall X_2 \Phi(X_1, X_2)$

- Level 3

- $\forall X_1 \exists X_2 \forall X_3 \Phi(X_1, X_2, X_3), \exists X_1 \forall X_2 \exists X_3 \Phi(X_1, X_2, X_3)$

Polynomial Space

- Quantified Boolean Expressions
 - $\exists X_1 \forall X_2 \exists X_3 \dots \exists X_{n-1} \forall X_n \Phi(X_1, X_2, X_3 \dots X_{n-1} X_n)$
- Space bounded games
 - Competitive Facility Location Problem
 - N x N Chess
- Counting problems
 - The number of Hamiltonian Circuits

