CSEP 521 Applied Algorithms

Richard Anderson Lecture 10 NP Completeness

Announcements

- Reading for this week
 - Chapter 8. NP-Completeness
- Final exam, March 18, 6:30 pm. At UW.
 - -2 hours
 - In class (CSE 303 / CSE 305)
 - Comprehensive
 - 67% post midterm, 33% pre midterm

Network Flow Summary









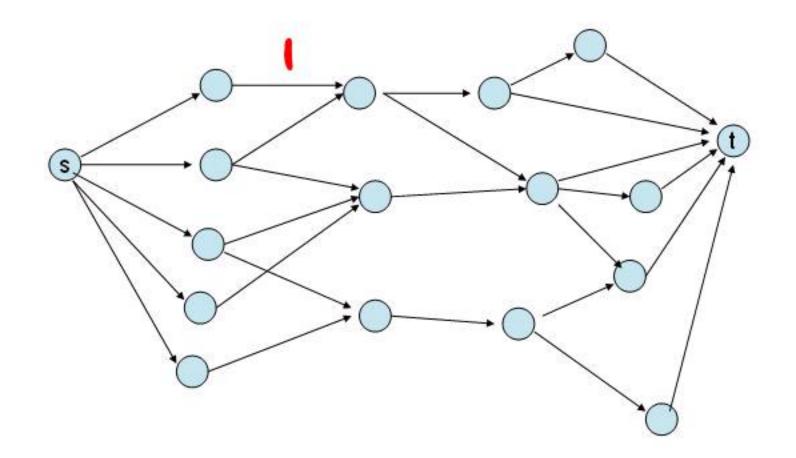
Network Flow

- Basic model
 - Graph with edge capacities, flow function, and conservation requirement
- Algorithmic approach
 - Residual Graph, Augmenting Paths, Ford-Fulkerson Algorithm
- Maxflow-MinCut Theorem
- Practical Algorithms: O(n³) or O(nmlog n)
 Blocking-Flow Algorithm

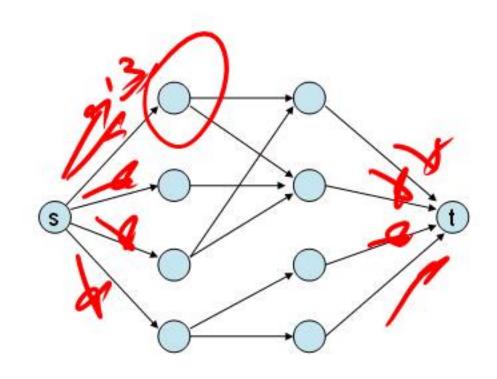
 - J Preflow-Push



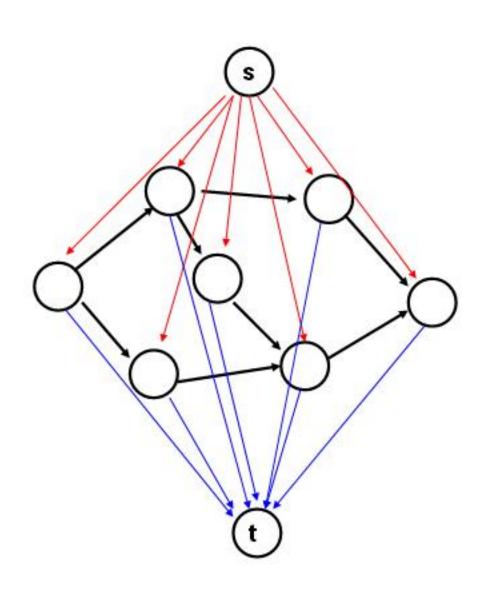
Variations on network flow



Resource Allocation Problems



Min-cut problems



NP Completeness

COMPUTERS, COMPLEXITY, AND INTRACTABILITY



I can't find an efficient algorithm, I guess I'm just too dumb.



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

Algorithms vs. Lower bounds

Algorithmic Theory

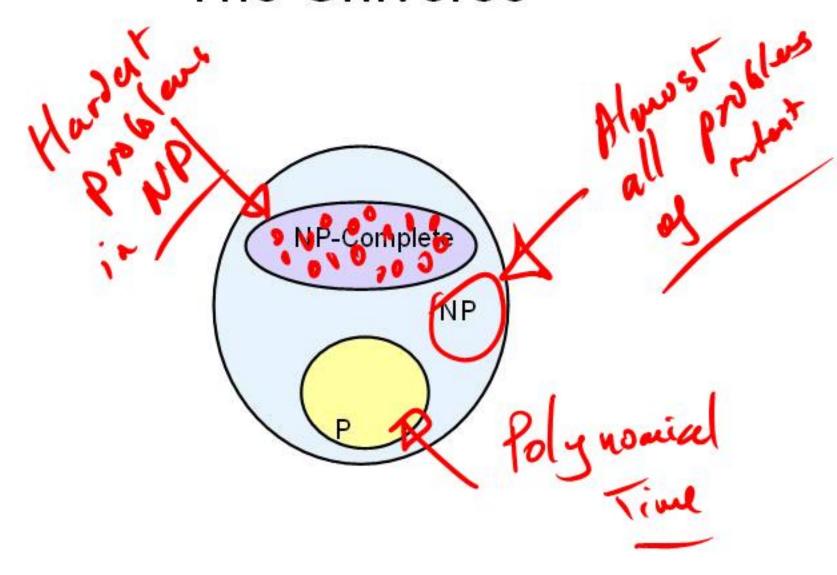
- What we can compute
 - I can solve problem X with resources R
- Proofs are almost always to give an algorithm that meets the resource bounds

Lower bounds

– How do we show that something can't be done?

Theory of NP Completeness

The Universe



Polynomial Time

- P: Class of problems that can be solved in polynomial time
 - Corresponds with problems that can be solved efficiently in practice
 - Right class to work with "theoretically"

Decision Problems

- Theory developed in terms of yes/no problems
 - Independent set
 - Given a graph G and an integer K, does G have an independent set of size at least K
 - Vertex cover
 - Given a graph G and an integer K, does the graph have a vertex cover of size at most K.

Definition of P

Decision problems for which there is a polynomial time algorithm

Problem	Description	Algorithm	Yes	No
MULTIPLE	Is x a multiple of y?	Grade school division	51,17	51, 16
RELPRIME	Are x and y relatively prime?	Euclid's algorithm	34,39	34,51
PRIMES	Is x prime?	Agrawal, Kayal, Saxena (2002)	53	51
EDIT- DISTANCE	Is the edit distance between x and y less than 5?	Dynamic programming	niether neither	acgggt ttttta
LSOLVE	Is there a vector x that satisfies Ax = b?	Gaussian elimination	0 1 1 4 2 4 -2 , 2 0 3 15 36	$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

What is NP?

Problems solvable in non-deterministic polynomial time . . .

 Problems where "yes" instances have polynomial time checkable certificates

50 > 51

Certificate examples

- Independent set of size K
 - The Independent Set
- Satifisfiable formula



- Truth assignment to the variables
- Hamiltonian Circuit Problem
 - A cycle including all of the vertices
- K-coloring a graph
 - Assignment of colors to the vertices

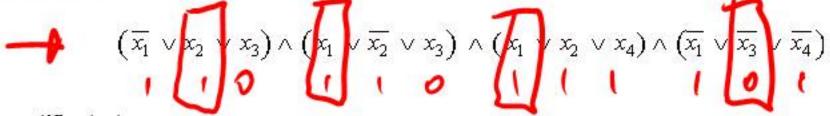
Certifiers and Certificates: 3-Satisfiability

SAT: Does a given CNF formula have a satisfying formula

Certificate: An assignment of truth values to the n boolean variables

Certifier: Check that each clause has at least one true literal,

instance s



certificate t



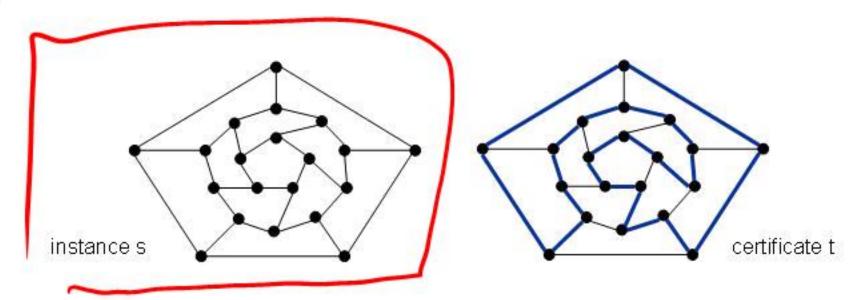
$$x_1 = 1$$
, $x_2 = 1$, $x_3 = 0$, $x_4 = 1$

Certifiers and Certificates: Hamiltonian Cycle

HAM-CYCLE. Given an undirected graph G = (V, E), does there exist a simple cycle C that visits every node?

Certificate. A permutation of the n nodes.

Certifier. Check that the permutation contains each node in V exactly once, and that there is an edge between each pair of adjacent nodes in the permutation.



Polynomial time reductions

- Y is Polynomial Time Reducible to X
 - Solve problem Y with a polynomial number of computation steps and a polynomial number of calls to a black box that solves X
 - Notations: Y <_P X

Lemma

 Suppose Y <_P X. If X can be solved in polynomial time, then Y can be solved in polynomial time.

Lemma

 Suppose Y <_P X. If Y cannot be solved in polynomial time, then X cannot be solved in polynomial time.

NP-Completeness

- A problem X is NP-complete if
 - X is in NP
 - For every Y in NP, Y < X



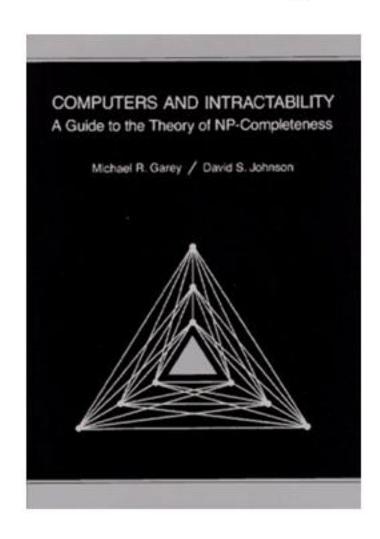


- X is a "hardest" problem in NP
- If X is NP-Complete, Z is in NP and X <_P Z
 - Then Z is NP-Complete

Cook's Theorem

 The Circuit Satisfiability Problem is NP-Complete

Garey and Johnson



History

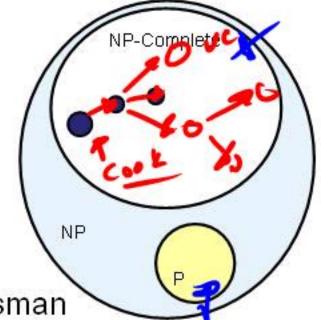
NK COM

- Jack Edmonds
 - Identified NP
- Steve Cook
 - Cook's Theorem NP-Completeness
- Dick Karp
 - Identified "standard" collection of NP-Complete Problems
- Leonid Levin
 - Independent discovery of NP-Completeness in USSR

Populating the NP-Completeness

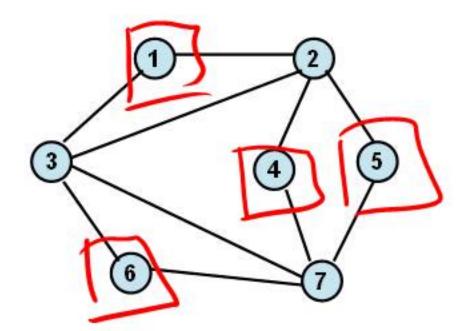
Universe

- Circuit Sat <_P 3-SAT
- 3-SAT <_□ Independent Set
- 3-SAT <_P Vertex Cover
- Independent Set
- 3-SAT <_□ Hamiltonian Circuit
- Hamiltonian Circuit
- 3-SAT <_P Integer Linear Programming
- 3-SAT <_P Graph Coloring
- 3-SAT <_P Subset Sum
- Subset Sum <_P Scheduling with Release times and deadlines



Sample Problems

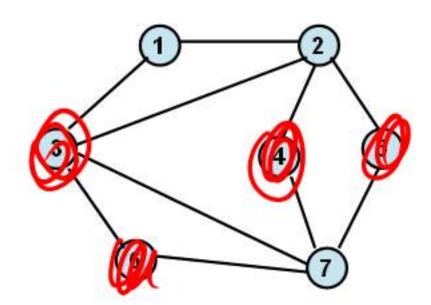
- Independent Set
 - Graph G = (V, E), a subset S of the vertices is independent if there are no edges between vertices in S



Vertex Cover

Vertex Cover

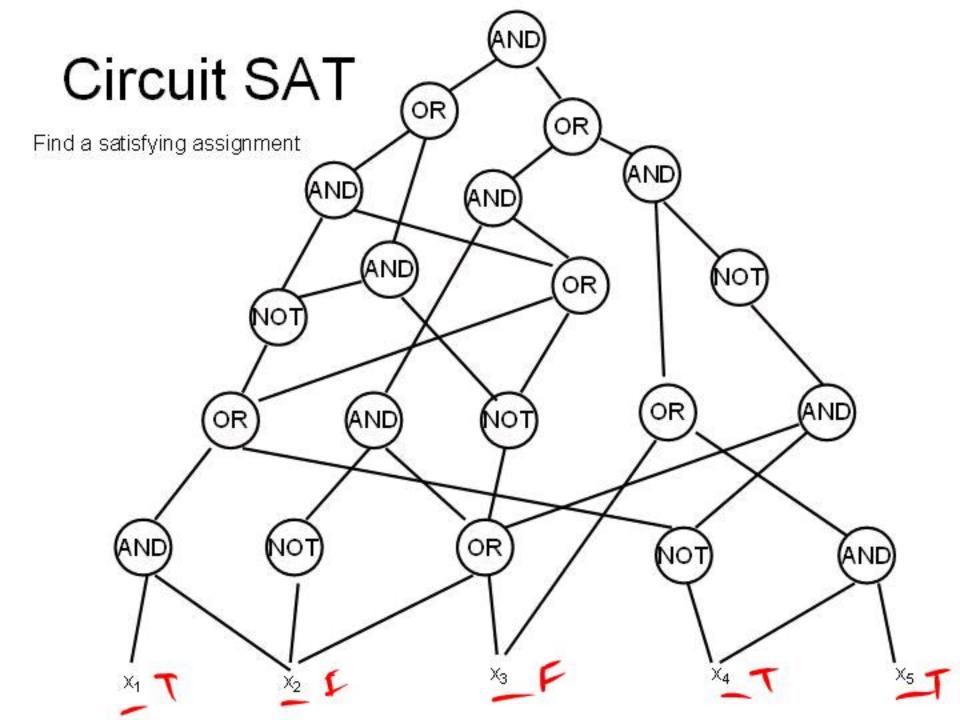
 Graph G = (V, E), a subset S of the vertices is a vertex cover if every edge in E has at least one endpoint in S



Cook's Theorem

 The Circuit Satisfiability Problem is NP-Complete

- Circuit Satisfiability
 - Given a boolean circuit, determine if there is an assignment of boolean values to the input to make the output true



Proof of Cook's Theorem

- Reduce an arbitrary problem Y in NP to X
- Let A be a non-deterministic polynomial time algorithm for Y
- Convert A to a circuit, so that Y is a Yes instance iff and only if the circuit is satisfiable

Populating the NP-Completeness

Universe

NP-Complete

NP

- Circuit Sat <_P 3-SAT
- 3-SAT <_□ Independent Set
- 3-SAT <_P Vertex Cover
- Independent Set
- 3-SAT <_P Hamiltonian Circuit
- Hamiltonian Circuit
- 3-SAT <_P Integer Linear Programming
- 3-SAT <_P Graph Coloring
- 3-SAT <_P Subset Sum
- Subset Sum <_P Scheduling with Release times and deadlines

Satisfiability

Literal: A Boolean variable or its negation.

$$x_i$$
 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: $(\overline{x_1} \lor x_2 \lor x_3) \land (x_1 \lor \overline{x_2} \lor x_3) \land (x_2 \lor x_3) \land (\overline{x_1} \lor \overline{x_2} \lor \overline{x_3})$

Yes: $x_1 = true$, $x_2 = true$ $x_3 = false$.

Circuit => Formula

3-SAT is NP-Complete

Theorem. 3-SAT is NP-complete.

Pf. Suffices to show that CIRCUIT-SAT ≤_p 3-SAT since 3-SAT is in NP.

- Let K be any circuit.
- Create a 3-SAT variable x_i for each circuit element i.
- Make circuit compute correct values at each node:

$$x_2 \vee x_3$$
, $\overline{x_2} \vee \overline{x_3}$

$$x_1 \vee \overline{x_4}, x_1 \vee \overline{x_5}, \overline{x_1} \vee x_4 \vee x_5$$

$$x_0 = x_1 \wedge x_2 \Rightarrow \text{ add 3 clauses:}$$

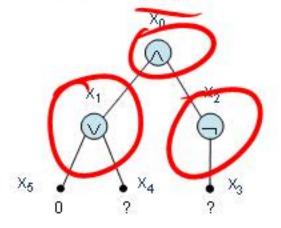
$$\overline{x_0} \lor x_1, \ x_0 \lor x_2, \ x_0 \lor \overline{x_1} \lor \overline{x_2}$$

Hard-coded input values and output value.

•
$$x_5 = 0 \Rightarrow \text{ add 1 clause: } \overline{x_5}$$

•
$$x_0 = 1 \Rightarrow \text{add 1 clause}$$
: x_0

Final step: turn clauses of length < 3 into clauses of length exactly 3.



output



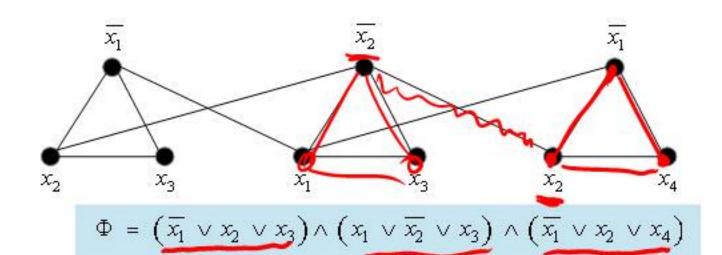
3 Satisfiability Reduces to Independent Set

Claim. 3-SAT ≤ p INDEPENDENT-SET.

Pf. Given an instance Φ of 3-SAT, we construct an instance (G, k) of INDEPENDENT-SET that has an independent set of size k iff Φ is satisfiable.

Construction.

- G contains 3 vertices for each clause, one for each literal.
- Connect 3 literals in a clause in a triangle.
- Connect literal to each of its negations.



G



3 Satisfiability Reduces to Independent Set

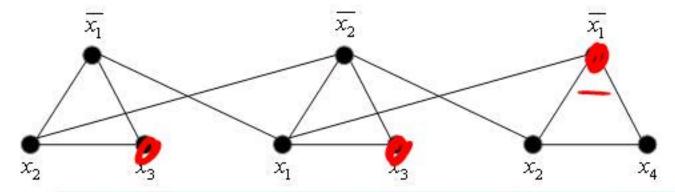
Claim. G contains independent set of size $k = |\Phi|$ iff Φ is satisfiable.

Pf. ⇒ Let S be independent set of size k.

- S must contain exactly one vertex in each triangle.
- Set these literals to true.
 ← and any other variables in a consistent way
- Truth assignment is consistent and all clauses are satisfied.

Pf ← Given satisfying assignment, select one true literal from each triangle. This is an independent set of size k. •





$$\Phi = (\overline{x_1} \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee x_4)$$

 Lemma: A set S is independent iff V-S is a vertex cover

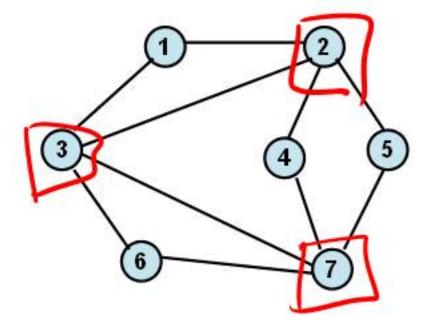
 To reduce IS to VC, we show that we can determine if a graph has an independent set of size K by testing for a Vertex cover of size n - K

IS <_P VC

Find a maximum independent set S

3 4 5

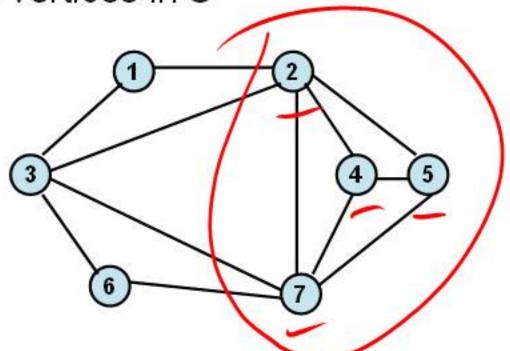
Show that V-S is a vertex cover



Clique

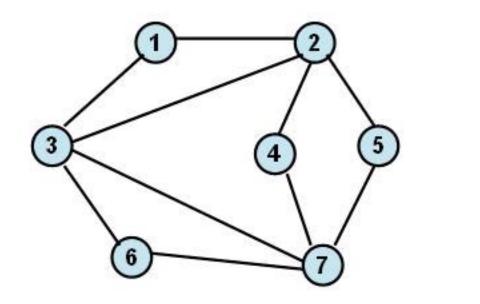
Clique

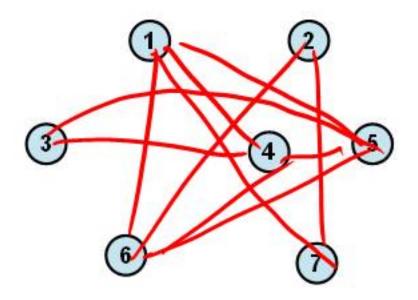
 Graph G = (V, E), a subset S of the vertices is a clique if there is an edge between every pair of vertices in S



Complement of a Graph

 Defn: G'=(V,E') is the complement of G=(V,E) if (u,v) is in E' iff (u,v) is not in E





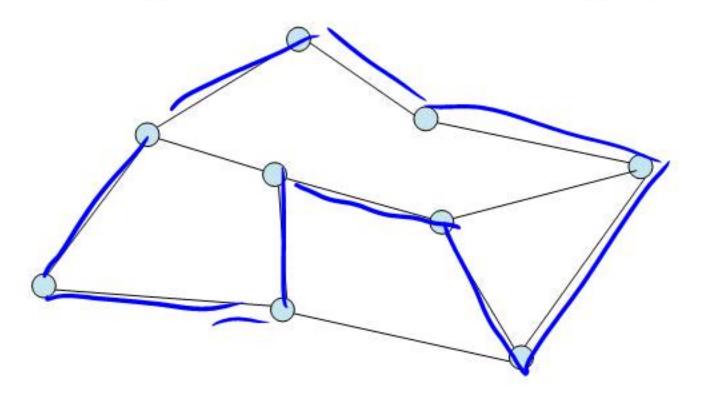
IS <_P Clique

 Lemma: S is Independent in G iff S is a Clique in the complement of G

 To reduce IS to Clique, we compute the complement of the graph. The complement has a clique of size K iff the original graph has an independent set of size K

Hamiltonian Circuit Problem

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

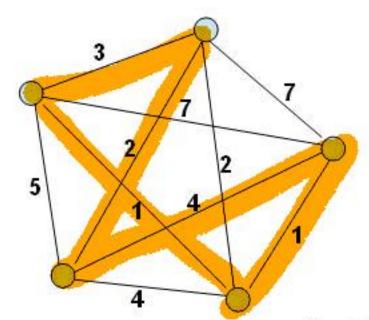


Thm: Hamiltonian Circuit is NP Complete

Reduction from 3-SAT

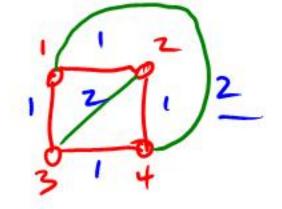
Traveling Salesman Problem

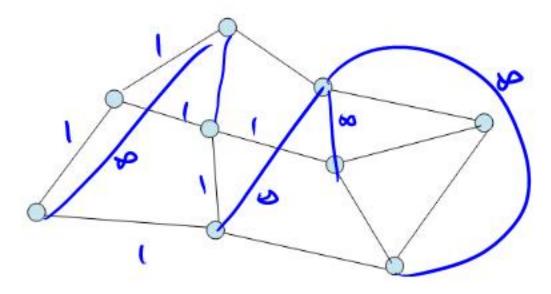
 Given a complete graph with edge weights, determine the shortest tour that includes all of the vertices (visit each vertex exactly once, and get back to the starting point)



Find the minimum cost tour

Thm: HC < TSP



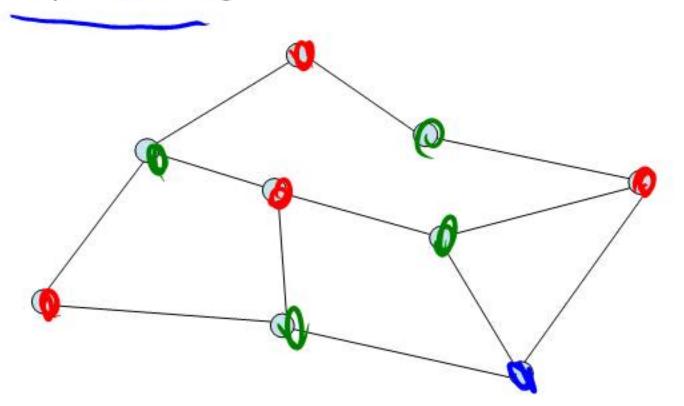


HC => TSP length

Graph Coloring

- NP-Complete
 - Graph K-coloring
 - Graph 3-coloring

- Polynomial
 - Graph 2-Coloring



Number Problems

- Subset sum problem
 - Given natural numbers w₁,..., w_n and a target number W, is there a subset that adds up to exactly W?

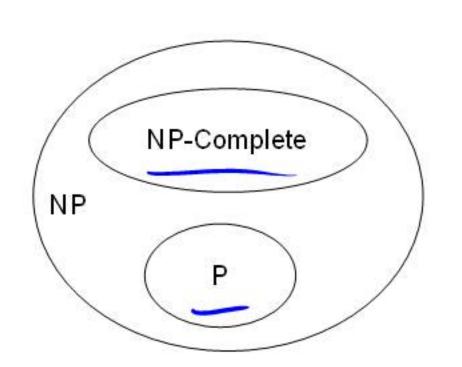
- Subset sum problem is NP-Complete
- Subset Sum problem can be solved in O(nW) time

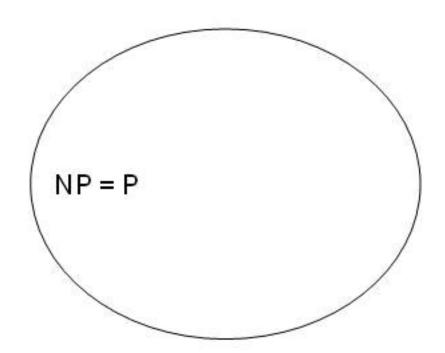
Subset sum problem

- The reduction to show Subset Sum is NPcomplete involves numbers with n digits
- In that case, the O(nW) algorithm is an exponential time and space algorithm

What we don't know

P vs. NP





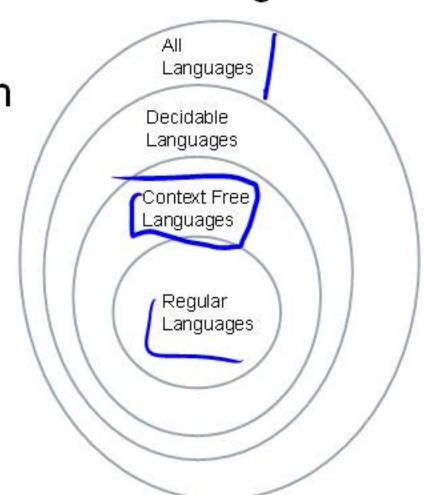
Complexity Theory

Computational requirements to recognize

languages

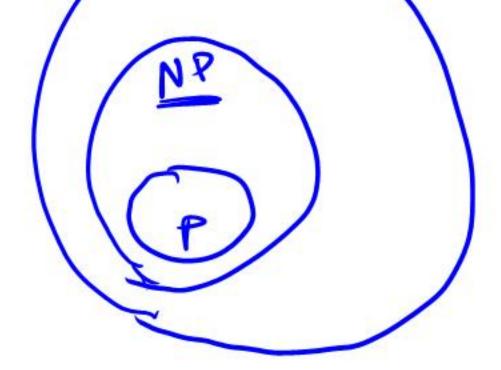
Models of Computation

- Resources
- Hierarchies



Time complexity

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

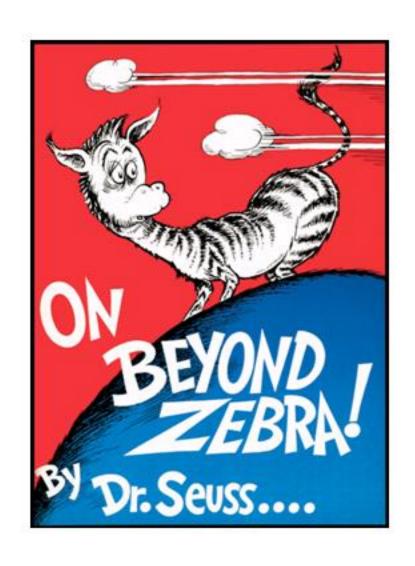


Space Complexity

- Amount of Space (Exclusive of Input)
- L: Logspace, problems that can be solved in O(log n) space for input of size n

 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 a 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

- Counting problems
 - The number of Hamiltonian Circuits in a graph