

Lecture27



CSE 417

Algorithms and Complexity

Autumn 2023

Lecture 27

Network Flow Applications

NP-Completeness

Announcements

- Homework 9
- Exam practice problems on course homepage
- Final Exam: Monday, December 11, 8:30 AM
– One Hour Fifty Minutes

Fri, Dec 1	Net Flow Applications
Mon, Dec 4	Net Flow Applications + NP-Completeness
Wed, Dec 6	NP-Completeness
Fri, Dec 8	NP-Completeness
Mon, Dec 11	Final Exam

Problem Reduction

- Reduce Problem A to Problem B
 - Convert an instance of Problem A to an instance of Problem B
 - Use a solution of Problem B to get a solution to Problem A
- Practical
 - Use a program for Problem B to solve Problem A
- Theoretical
 - Show that Problem B is at least as hard as Problem A

Minimum Cut Applications

- Image Segmentation
- Open Pit Mining / Task Selection Problem
- Reduction to Min Cut problem

S, T is a cut if S, T is a partition of the vertices with s in S and t in T

The capacity of an S, T cut is the sum of the capacities of all edges going from S to T

Image Segmentation

- Separate foreground from background

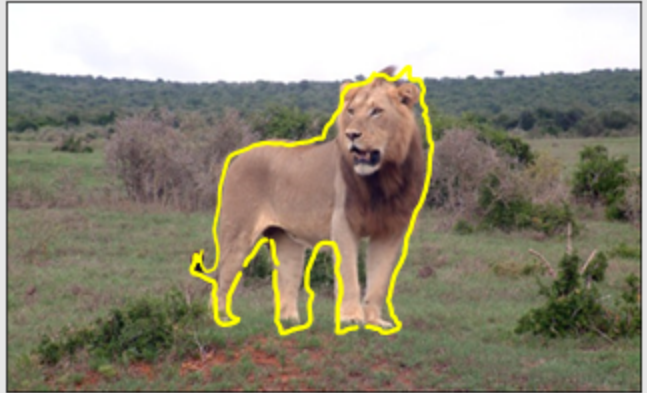
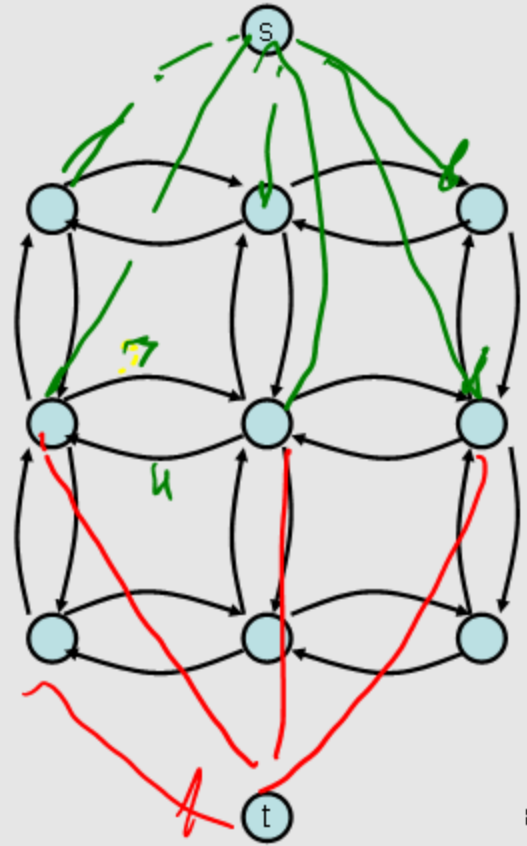
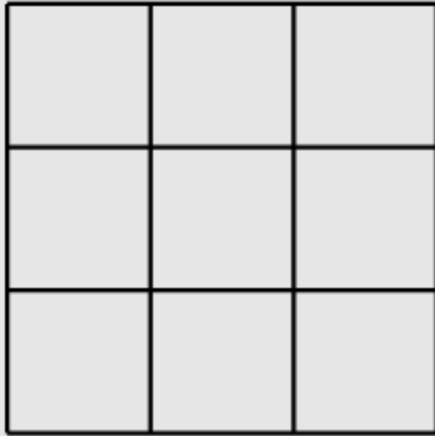




Image analysis

- a_i : value of assigning pixel i to the foreground
- b_j : value of assigning pixel j to the background
- p_{ij} : penalty for assigning i to the foreground, j to the background or vice versa
- A : foreground, B : background
- $Q(A, B) = \sum_{\{i \text{ in } A\}} a_i + \sum_{\{j \text{ in } B\}} b_j - \sum_{\{(i,j) \text{ in } E, i \text{ in } A, j \text{ in } B\}} p_{ij}$

Pixel graph to flow graph

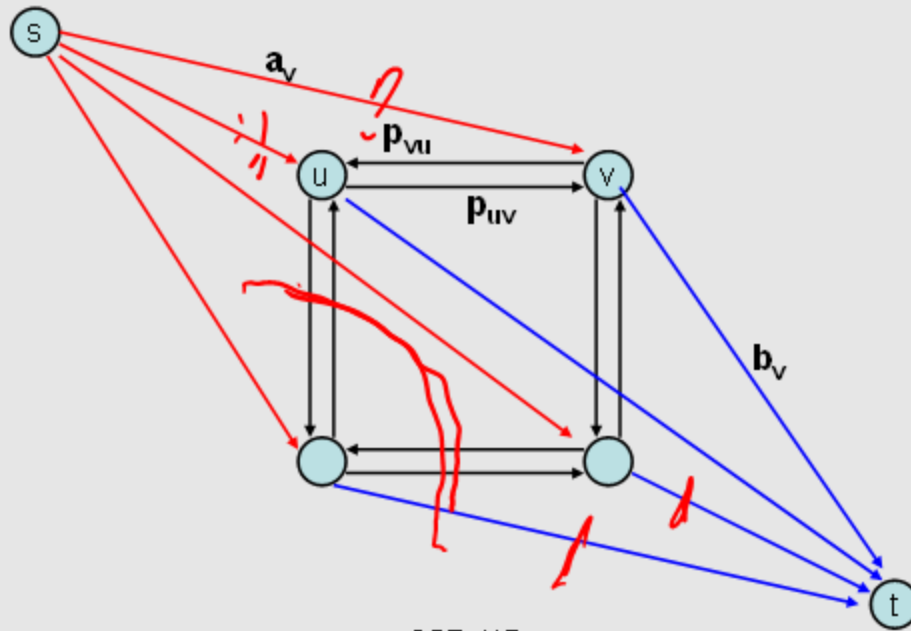


12/4/2023

CSE 417

8

Mincut Construction



12/4/2023

CSE 417

9

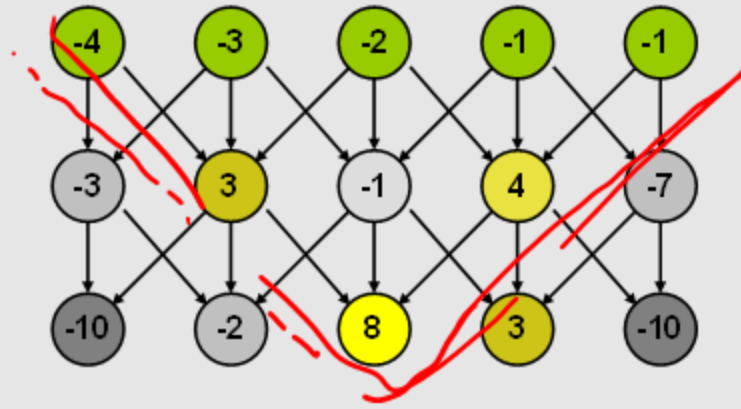
Open Pit Mining (Task selection)



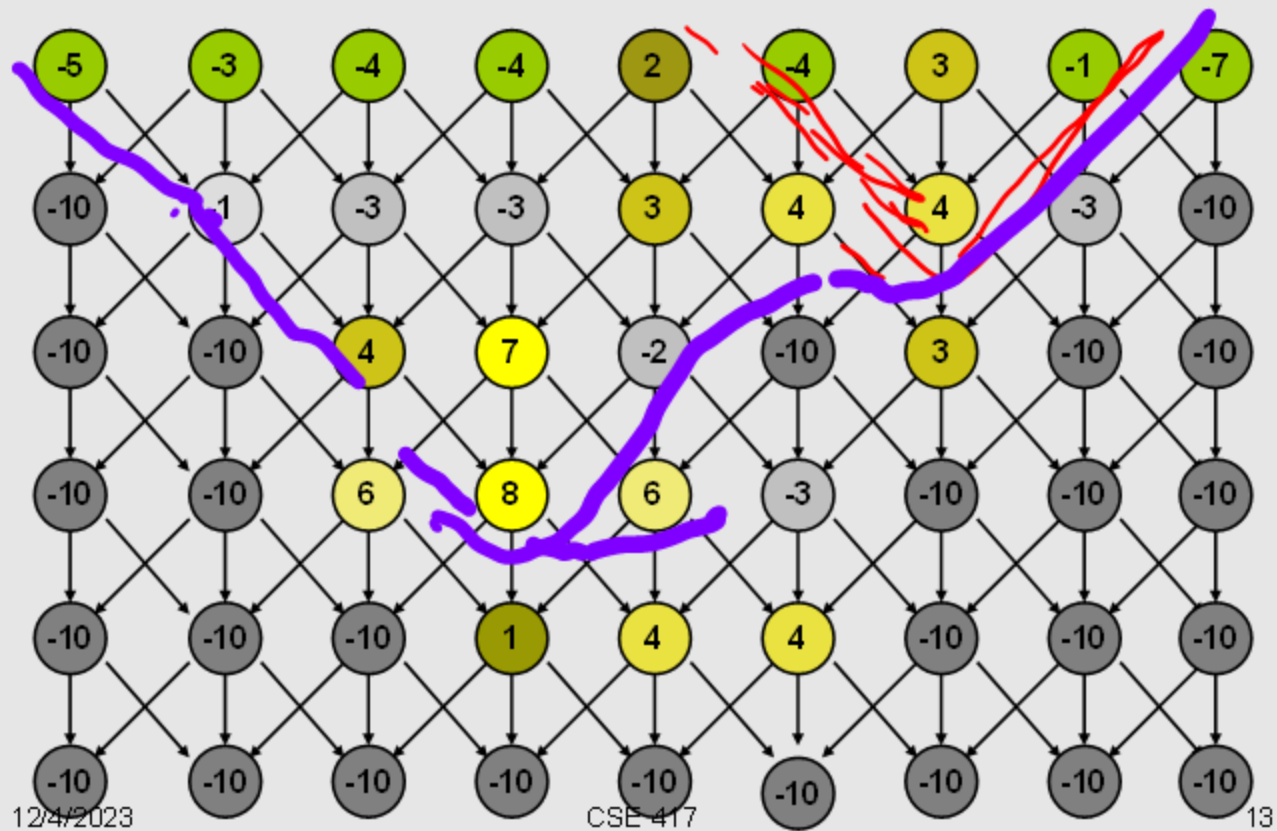
Open Pit Mining

- Each unit of earth has a profit (possibly negative)
- Getting to the ore below the surface requires removing the dirt above
- Test drilling gives reasonable estimates of costs
- Plan an optimal mining operation

Mine Graph



Determine an optimal mine



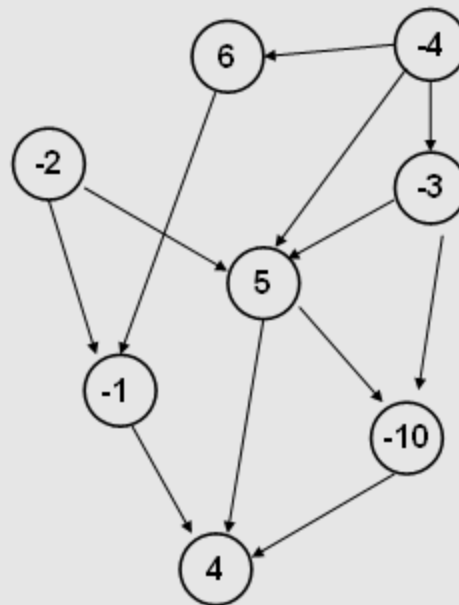
12/4/2023

CSE 417

13

Generalization

- Precedence graph $G=(V,E)$
- Each v in V has a profit $p(v)$
- A set F is *feasible* if when w in F , and (v,w) in E , then v in F .
- Find a feasible set to maximize the profit

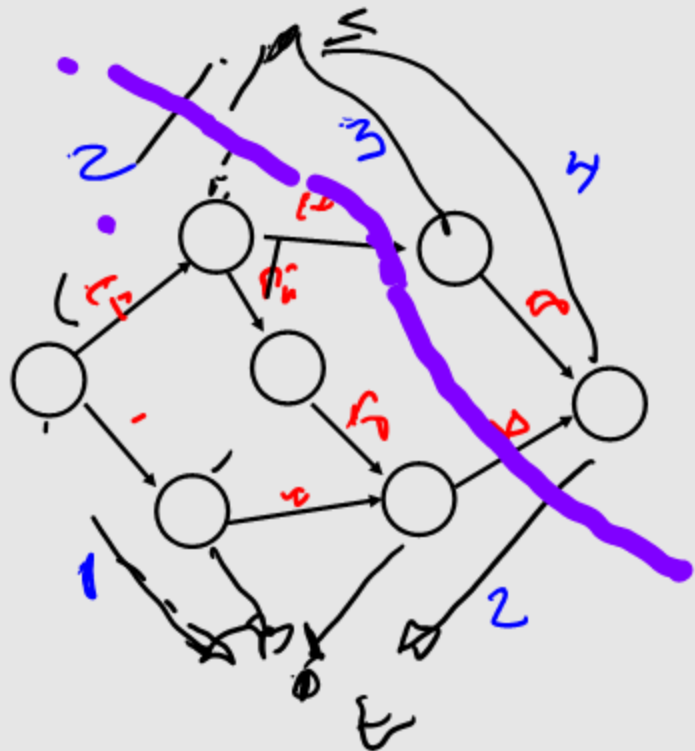


Min cut algorithm for profit maximization

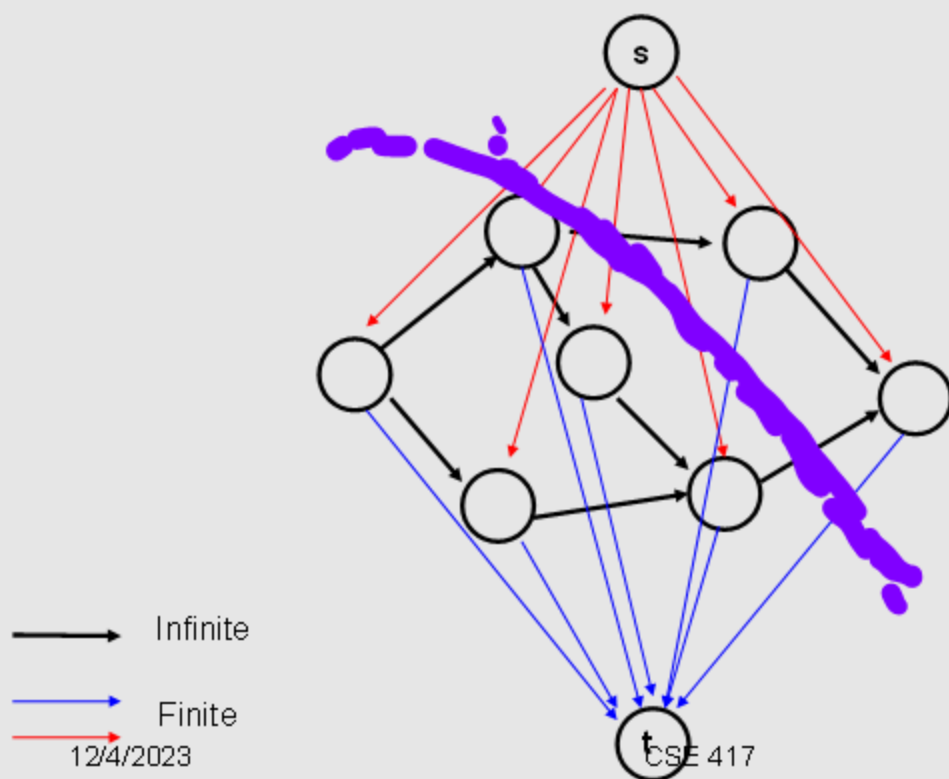
- Construct a flow graph where the minimum cut identifies a feasible set that maximizes profit

Precedence graph construction

- Precedence graph $G=(V,E)$
- Each edge in E has infinite capacity
- Add vertices s, t
- Each vertex in V is attached to s and t with finite capacity edges

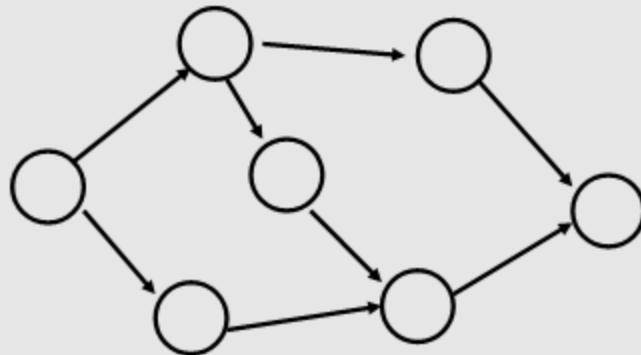


Find a **finite** value cut with at least two vertices on each side of the cut



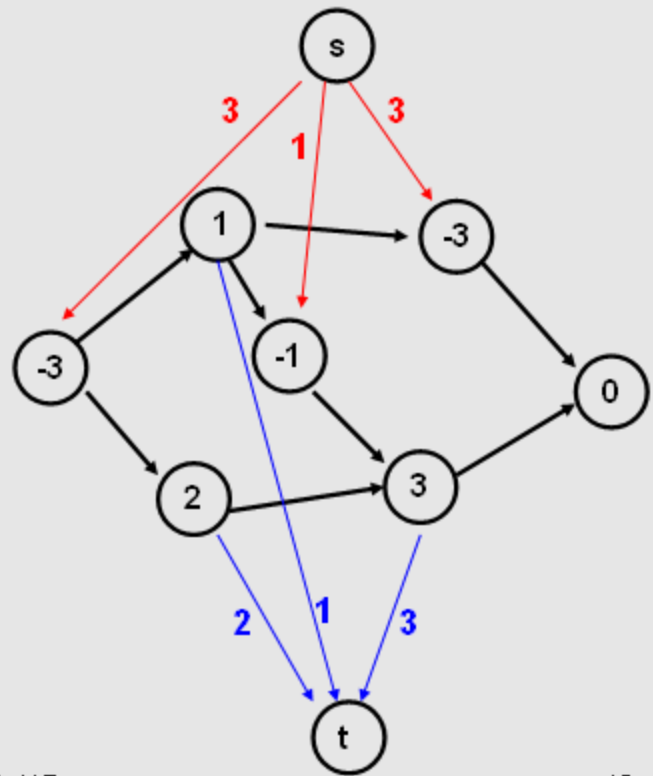
The sink side of a finite cut is a feasible set

- No edges permitted from S to T
- If a vertex is in T , all of its ancestors are in T



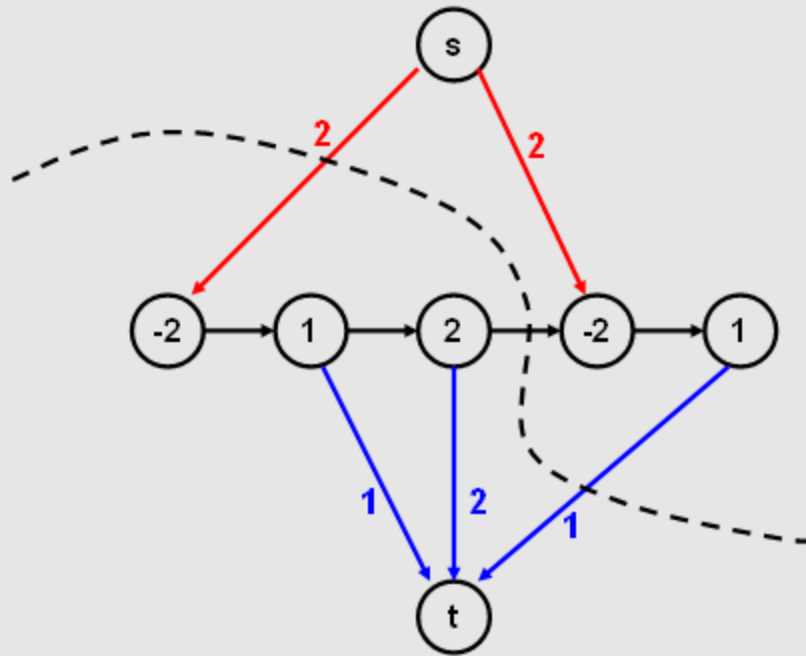
Setting the costs

- If $p(v) > 0$,
 - $\text{cap}(v,t) = p(v)$
 - $\text{cap}(s,v) = 0$
- If $p(v) < 0$
 - $\text{cap}(s,v) = -p(v)$
 - $\text{cap}(v,t) = 0$
- If $p(v) = 0$
 - $\text{cap}(s,v) = 0$
 - $\text{cap}(v,t) = 0$



Minimum cut gives optimal solution

Why?



12/4/2023

CSE 417

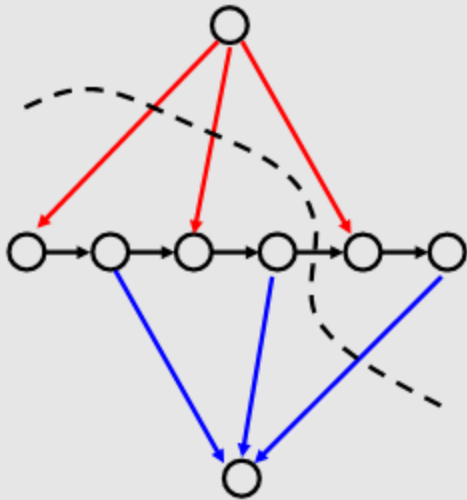
20

Computing the Profit

- $\text{Cost}(W) = \sum_{\{w \text{ in } W; p(w) < 0\}} -p(w)$
- $\text{Benefit}(W) = \sum_{\{w \text{ in } W; p(w) > 0\}} p(w)$
- $\text{Profit}(W) = \text{Benefit}(W) - \text{Cost}(W)$

- **Maximum cost and benefit**
 - $C = \text{Cost}(V)$
 - $B = \text{Benefit}(V)$

Express $\text{Cap}(S,T)$ in terms of B , C , $\text{Cost}(T)$, $\text{Benefit}(T)$, and $\text{Profit}(T)$



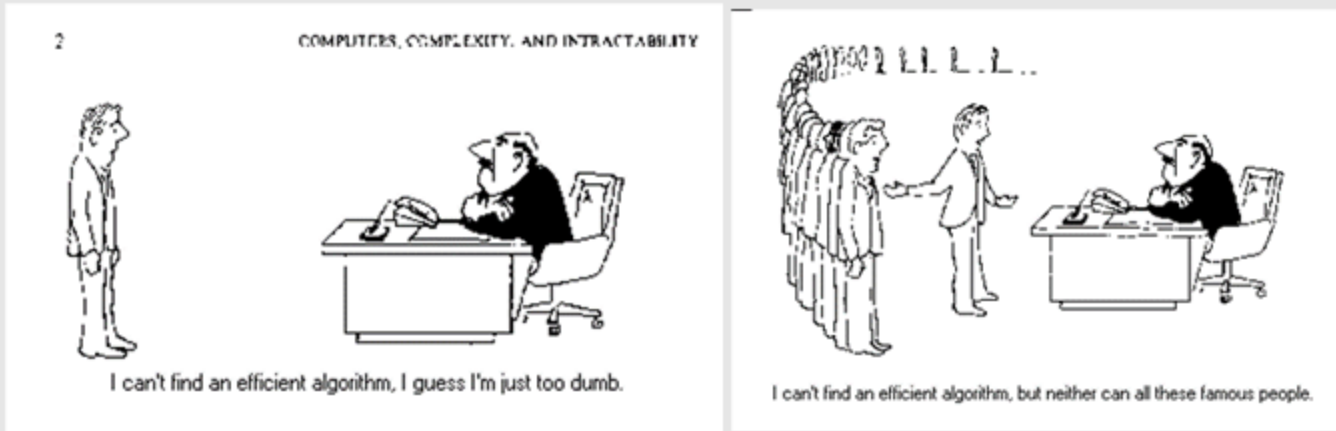
$$\begin{aligned} \text{Cap}(S,T) &= \text{Cost}(T) + \text{Ben}(S) = \text{Cost}(T) + \text{Ben}(S) + \text{Ben}(T) - \text{Ben}(T) \\ &= B + \text{Cost}(T) - \text{Ben}(T) = B - \text{Profit}(T) \end{aligned}$$



NP-Completeness



NP Completeness



Algorithms vs. Lower bounds

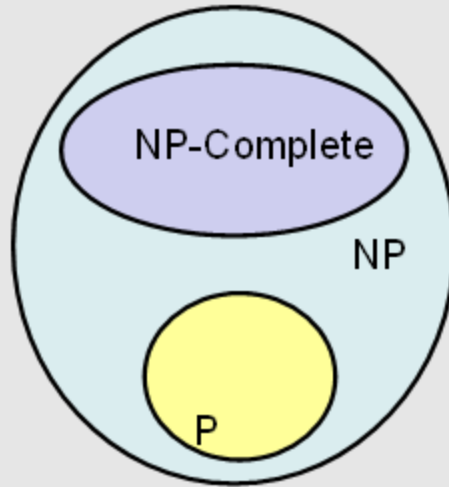
- **Algorithmic Theory**
 - What we can compute
 - I can solve problem X with resources R
 - Proofs are almost always to give an algorithm that meets the resource bounds
- **Lower bounds**
 - How do we show that something can't be done?

Theory of NP Completeness

P vs. NP

$P \neq NP$

The Universe



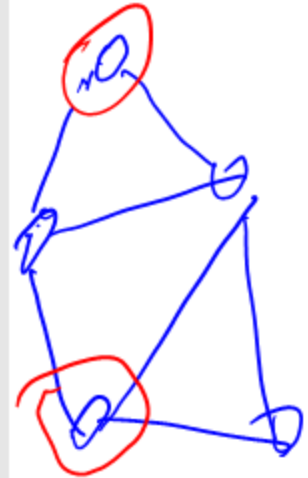
$P = NP$

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
 - Shortest Path
 - Given a graph G with edge lengths, a start vertex s , and end vertex t , and an integer K , does the graph have a path between s and t of length at most K



What is NP?

- Problems solvable in non-deterministic polynomial time . . .
- Problems where “yes” instances have polynomial time checkable certificates

Certificate examples

- Independent set of size K
 - The Independent Set
- Satisfiable 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

$x \wedge y \vee z$

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

$$(\bar{x}_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee \bar{x}_4)$$

certificate t

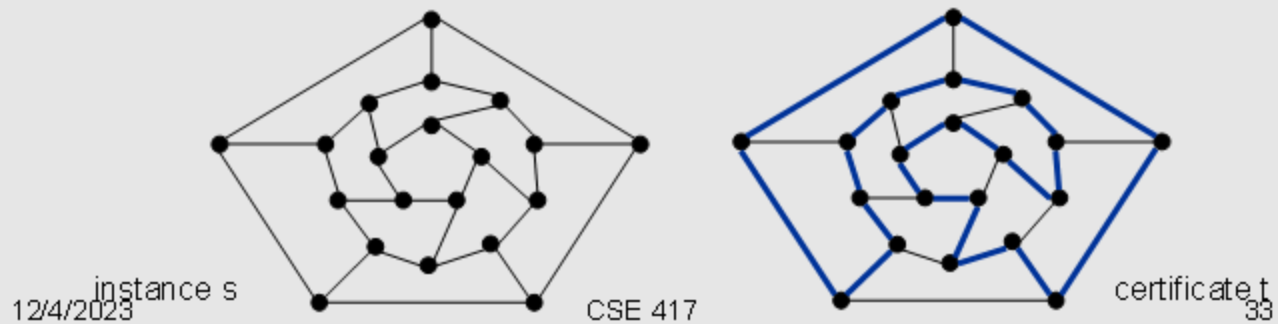
$$x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 1$$

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: $Y <_p X$
- Usually, this is converting an input of Y to an input for X, solving X, and then converting the answer back

Composability Lemma

- If $X <_p Y$ and $Y <_p Z$ then $X <_p Z$

Lemmas

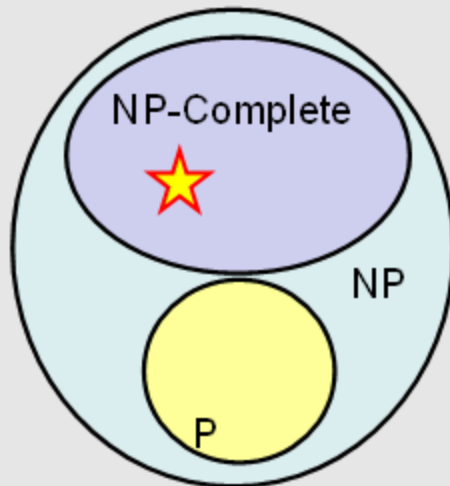
- Suppose $Y <_p X$. If X can be solved in polynomial time, then Y can be solved in polynomial time.
- Suppose $Y <_p X$. If Y cannot be solved in polynomial time, then X cannot be solved in polynomial time.

NP-Completeness

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

- There is an NP Complete problem
 - The Circuit Satisfiability Problem



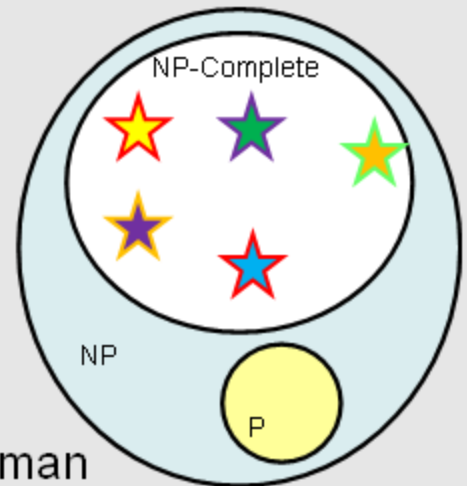
12/4/2023

CSE 417

38

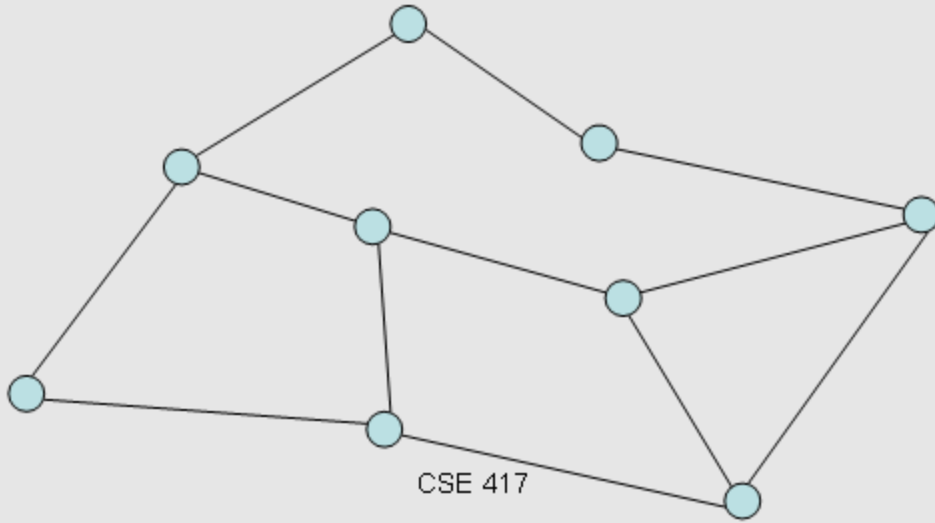
Populating the NP-Completeness Universe

- Circuit Sat $<_P$ 3-SAT
- 3-SAT $<_P$ Independent Set
- 3-SAT $<_P$ Vertex Cover
- Independent Set $<_P$ Clique
- 3-SAT $<_P$ Hamiltonian Circuit
- Hamiltonian Circuit $<_P$ Traveling Salesman
- 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



Graph Coloring

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



12/4/2023

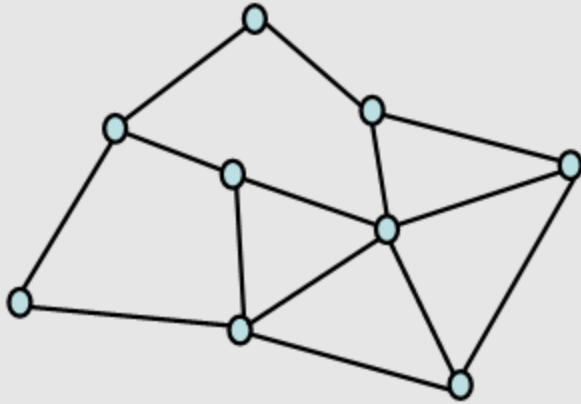
CSE 417

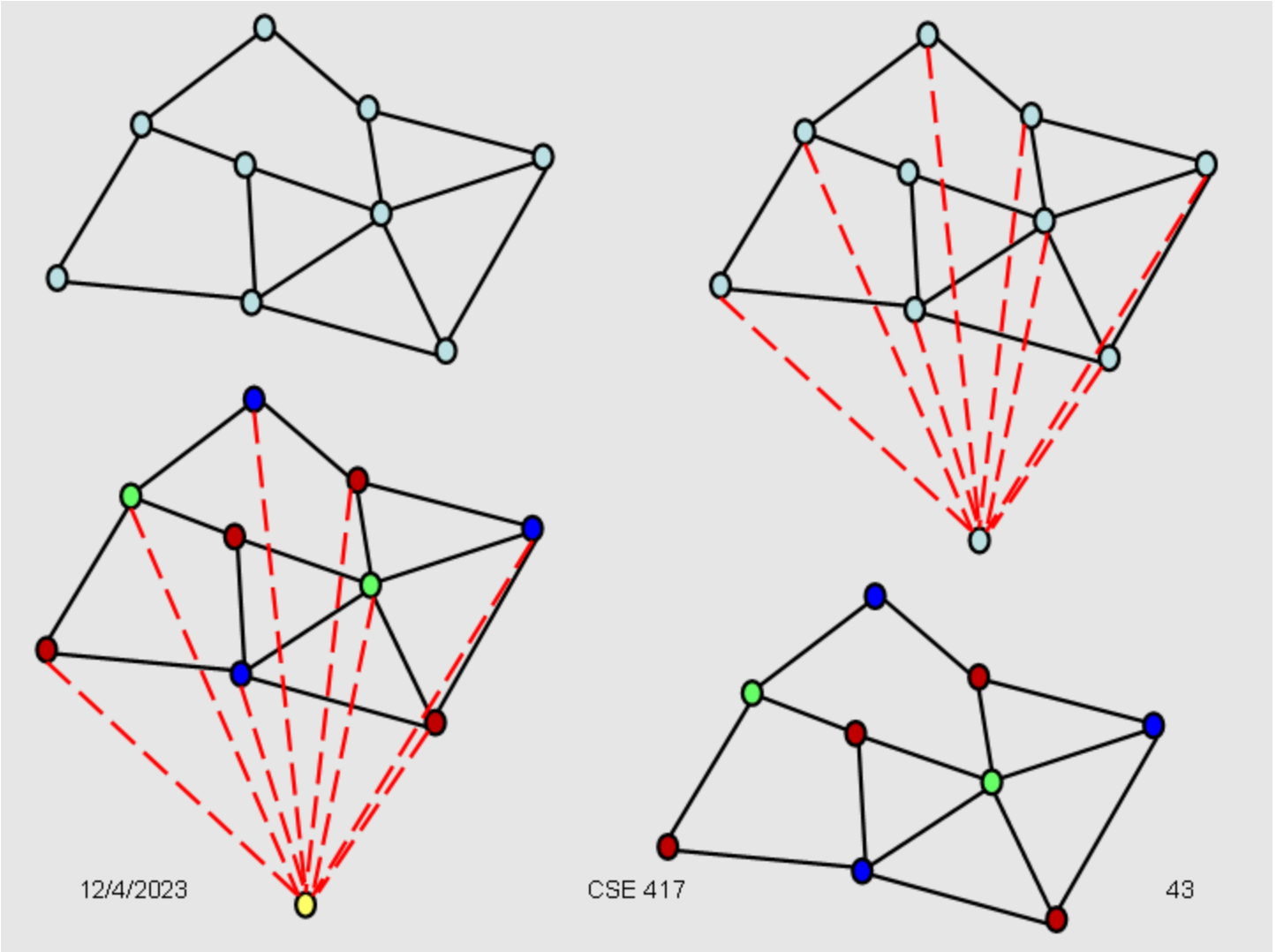
40

Graph 4-Coloring

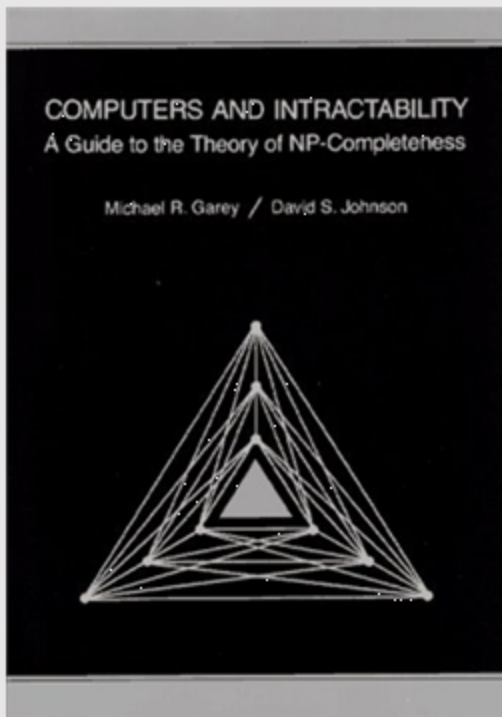
- Given a graph G , can G be colored with 4 colors?
- Prove 4-Coloring is NP Complete
- Proof: 3-Coloring $<_p$ 4-Coloring
- Show that you can 3-Color a graph if you have an algorithm to 4-Color a graph

3-Coloring $<_p$ 4-Coloring





Garey and Johnson



12/4/2023

CSE 417

44

P vs. NP Question

