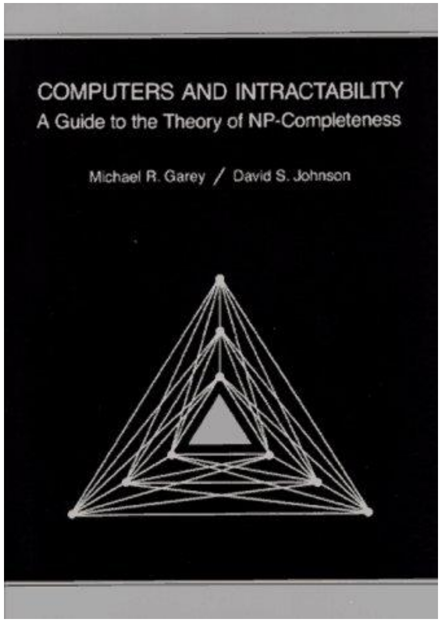


CSE 421 Algorithms

Richard Anderson

Lecture 25

Min Cut Applications and NP-Completeness

Today’s topics

- Min Cut Applications
- NP-Completeness

Application of Min-cut

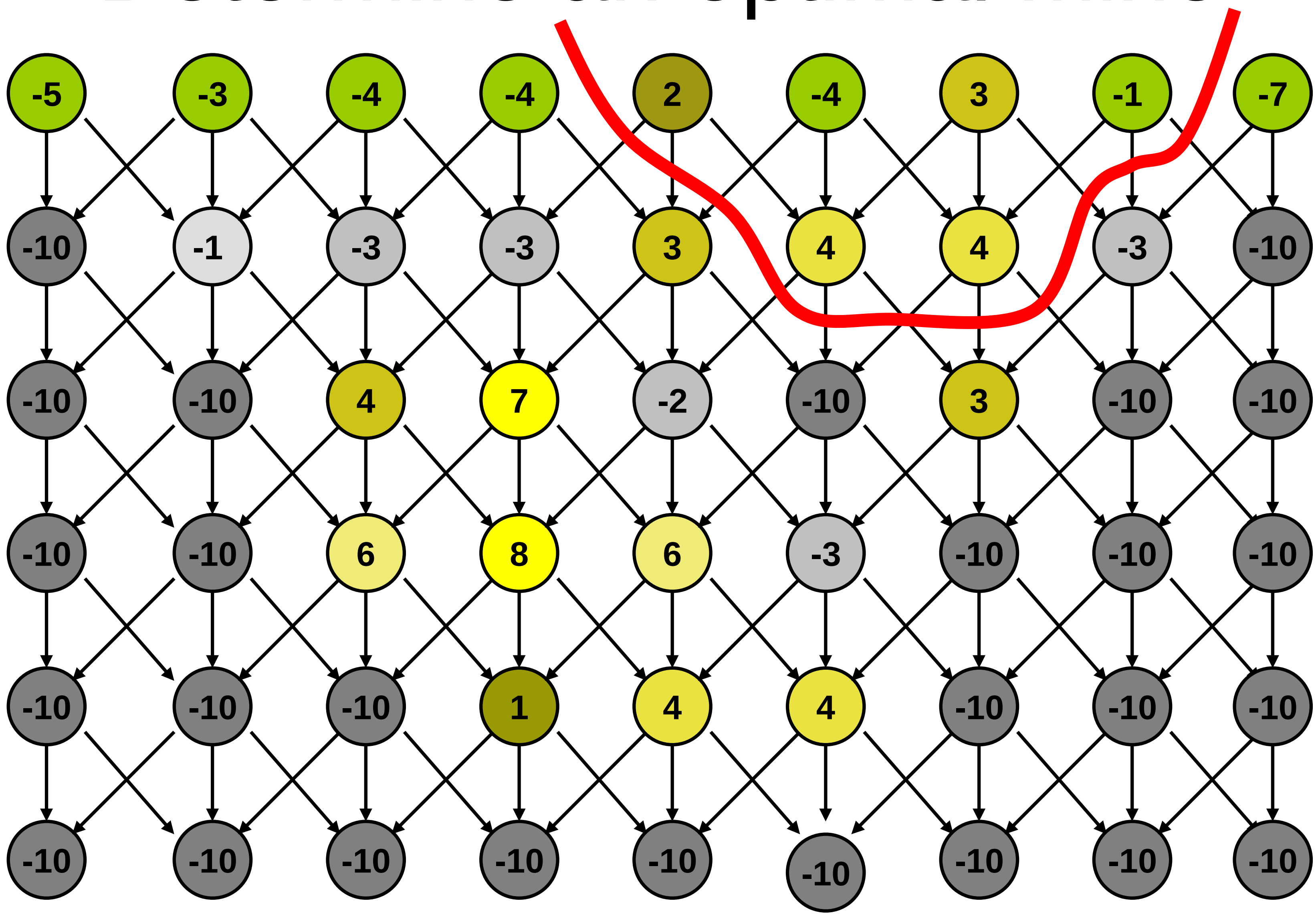
- Open Pit Mining Problem
- Task Selection Problem
- Reduction to Min Cut problem

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

Open Pit Mining

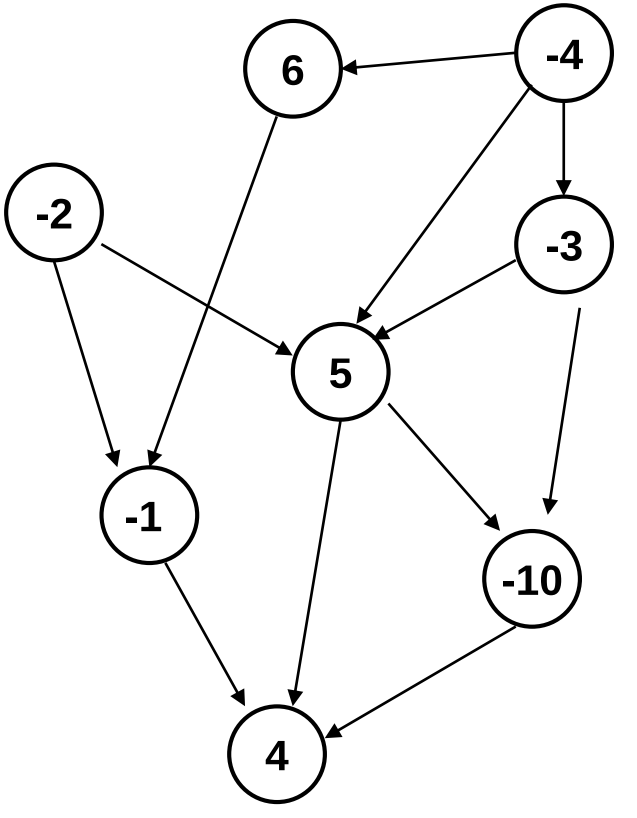
- Each unit of earth has a profit (possibly negative)
- Getting to the ore below the surface requires removing the dirt above
- Test drilling gives reasonable estimates of costs
- Plan an optimal mining operation

Determine an optimal mine



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

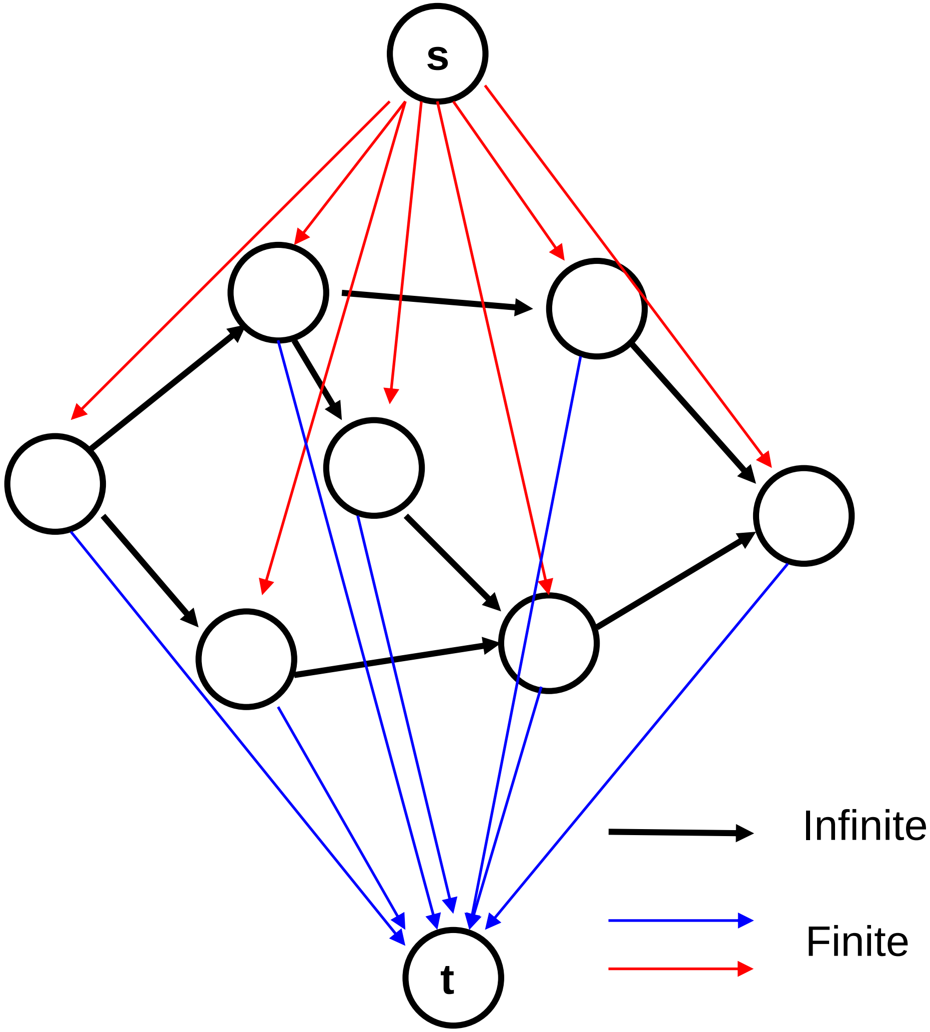


Min cut algorithm for profit maximization

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

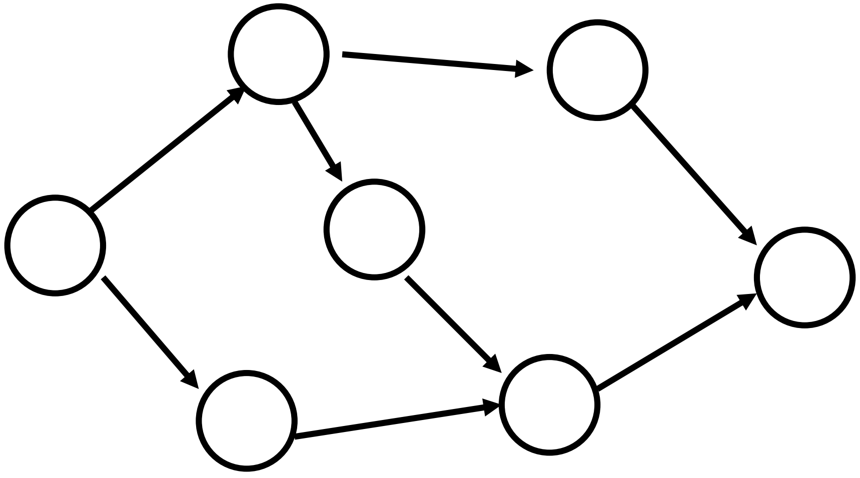
Find a finite value cut with at least two vertices on each side of the cut

- 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



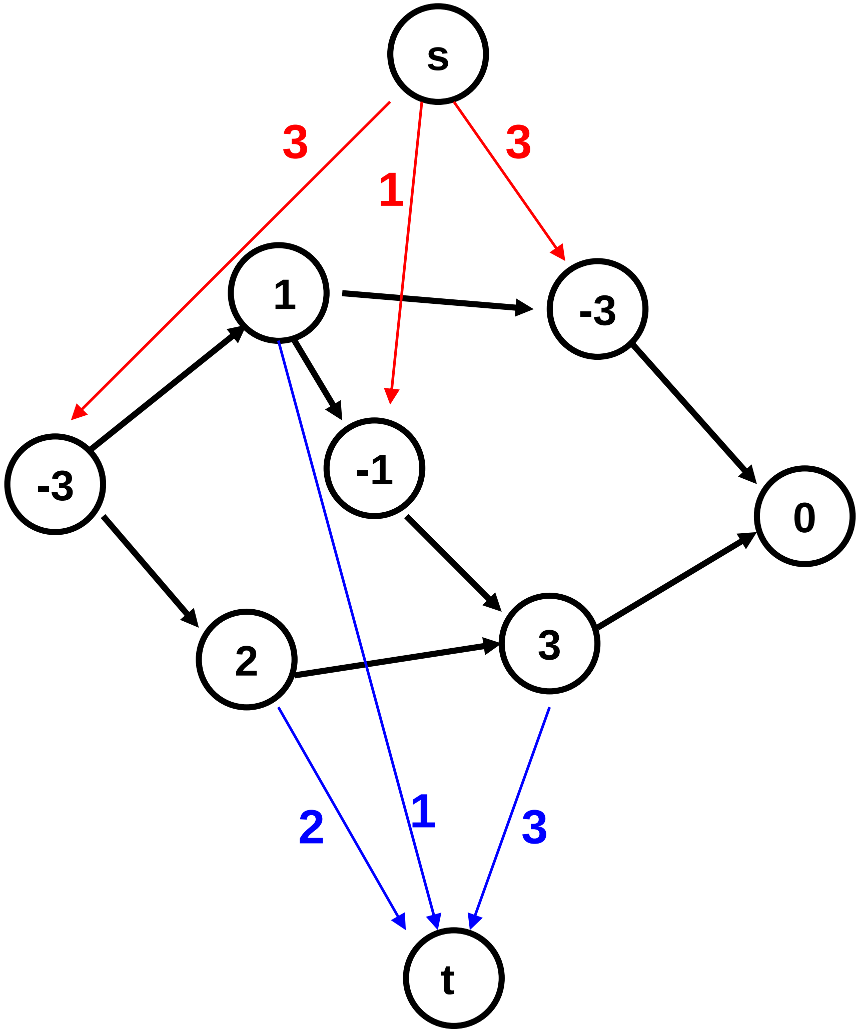
The sink side of a finite cut is a feasible set

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

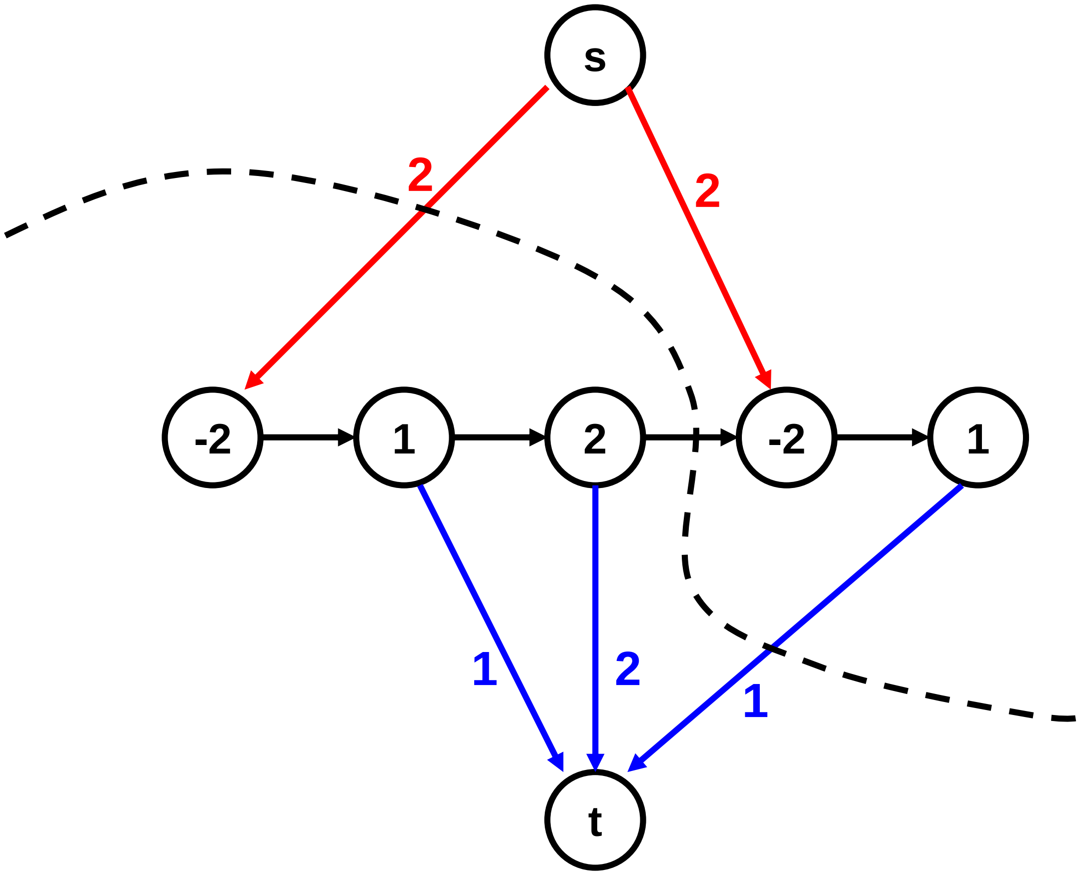


Setting the costs

- If p(v) > 0,
  - cap(v,t) = p(v)
  - cap(s,v) = 0
- If p(v) < 0
  - cap(s,v) = -p(v)
  - cap(v,t) = 0
- If p(v) = 0
  - cap(s,v) = 0
  - cap(v,t) = 0



Minimum cut gives optimal solution
Why?

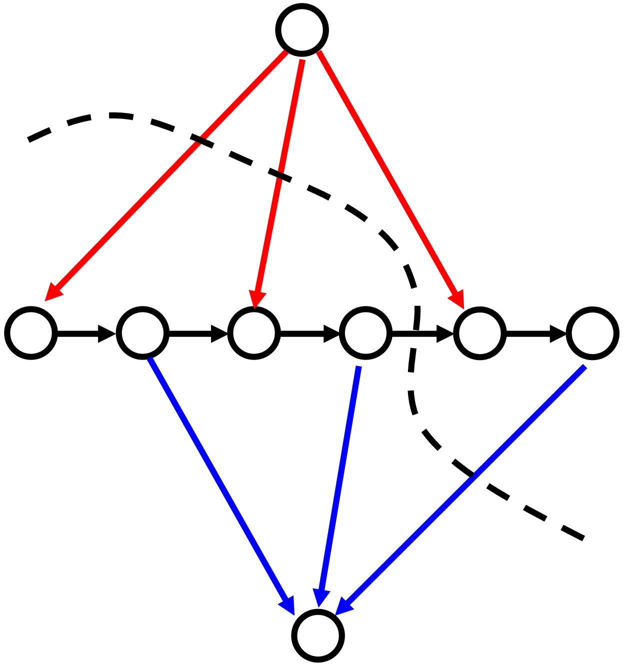


Computing the Profit

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

- Maximum cost and benefit
  - C = Cost(V)
  - B = Benefit(V)

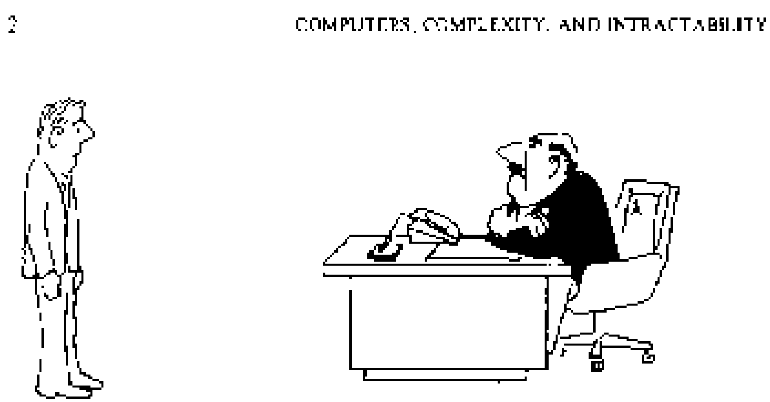
Express Cap(S,T) in terms of B, C, Cost(T), Benefit(T), and Profit(T)



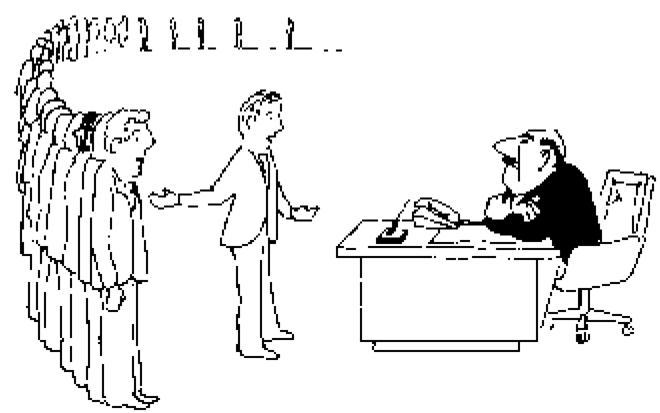
Cap(S,T) = Cost(T) + Ben(S) = Cost(T) + Ben(S) + Ben(T) – Ben(T)

= B + Cost(T) – Ben(T) = B – Profit(T)

NP Completeness



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



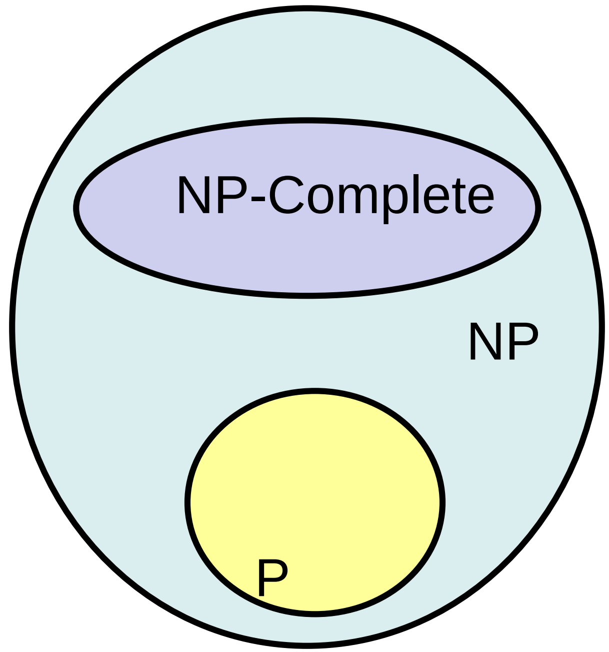
I can't find an efficient algorithm, but neither can all these famous people.

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	$\begin{bmatrix} 0 & 1 & 1 \\ 2 & 4 & -2 \\ 0 & 3 & 15 \end{bmatrix}, \begin{bmatrix} 4 \\ 2 \\ 36 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

What is NP?

- Problems solvable in non-deterministic polynomial time . . .

- Problems where “yes” instances have polynomial time checkable certificates

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

$$\left(\overline{x_1} \lor x_2 \lor x_3\right) \land \left(x_1 \lor \overline{x_2} \lor x_3\right) \land \left(x_1 \lor x_2 \lor x_4\right) \land \left(\overline{x_1} \lor \overline{x_3} \lor \overline{x_4}\right)$$

certificate t

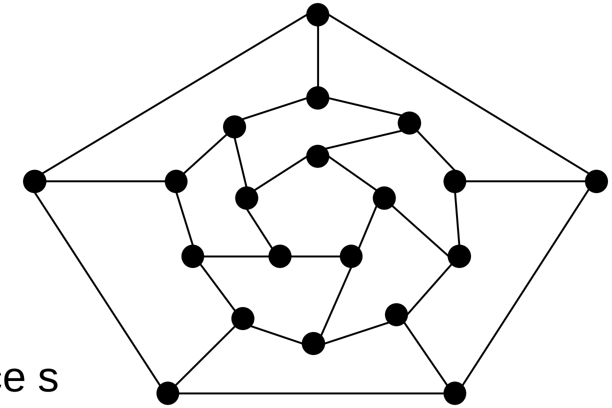
$$x_1 = 1,\ x_2 = 1,\ x_3 = 0,\ x_4 = 1$$

Certifiers and Certificates: Hamiltonian Cycle

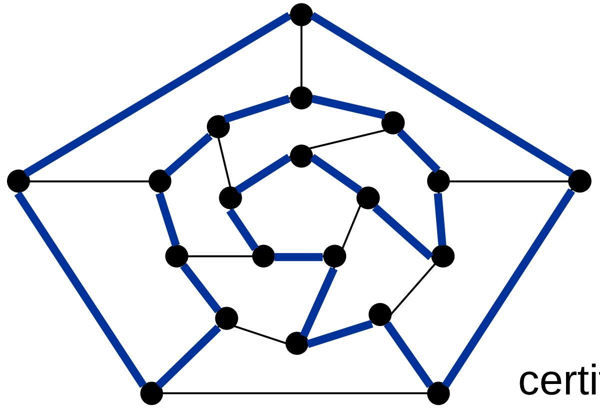
HAM-CYCLE. Given an undirected graph G = (V, E), does there exist a simple cycle C that visits every node?

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.



instance s



certificate t

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.

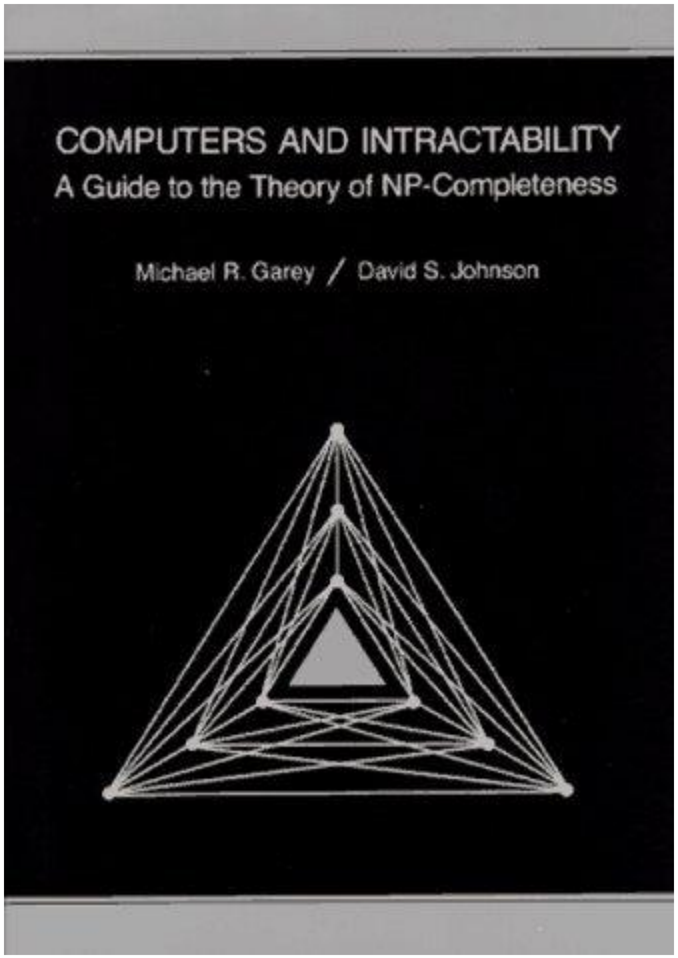
NP-Completeness

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

- The Circuit Satisfiability Problem is NP-Complete

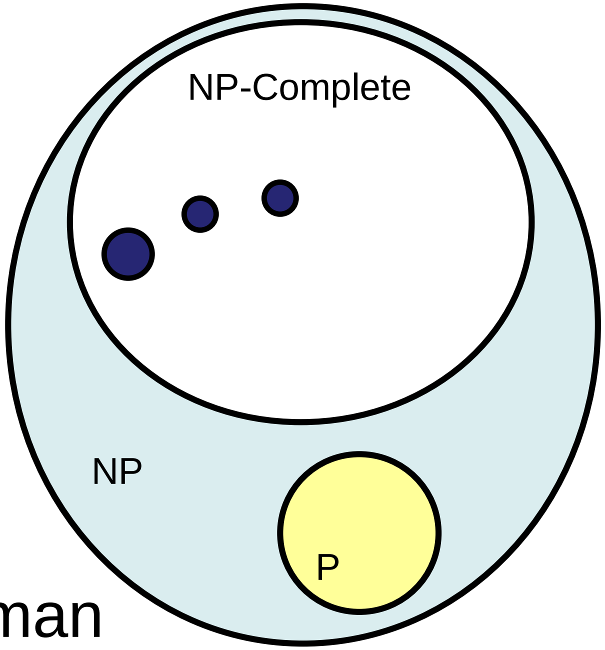
Garey and Johnson



History

- 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 $<_P$ 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