

Announcements

HW9 due Friday (approximation)

Section this week: quiz (on div&conquer) and practice approximation

Upcoming plans:

Today: ~~_____~~ *solving hard problems*

Monday: solving NP-hard problems exactly

Wed-Friday: using hardness for crypto (lectures by Noah Stephens-Davidowitz)

Next week: problems even harder than NP-complete

Bowers CIS Pre-Enroll Event



CornellBowers
College of Computing + Information Science



Open to all CS/IS/ISST pre-majors, majors, minors, and anyone interested in computing.

today
→ April 13, 2026
4:00 - 5:30 PM

CIS Building, Wayfair Commons (Room 132), Conine and Shah Families Active Learning Classroom (Room 142)

Join by Web PollEv.com/evatardos772



How you feel about the divide and conquer homework that was due this weekend?

- A. Easy enough, spent <3 hours
- B. OK, spent 3-6 hours
- C. Doable, spent 7-10 hours
- D. Too hard, did not solve, despite spending more than 7 hours
- E. Hard, managed to solve, but spent 11-15 hours
- F. Very hard, spent over 15 hours
- G. Did not have time to make a serious attempt at solving

Coping with NP-hard problems

Options

1. Solve exactly, hope won't take too long
2. approximate solution in polynomial time
3. try to find poly time algorithm

\$1M

2. approximation algorithms
last & kw9

today: bit of item 1.
today

→ - practical expectation / hope

- on random input prove bound expected time

All problems in NP solvable in ??

SAT: $\phi \sim (x_1 \vee \bar{x}_2 \vee x_3) \wedge \dots$

x_1, \dots, x_n variables, m clauses

brute force algorithm

try all options $x_1, \dots, x_n \in \{T, F\}$

2^n options

Exp = exponential time

$2^{O(n^c)}$

Claim $NP \subseteq EXP$

Proof 1:

A in NP, SAT NP-complete

$A \leq_p SAT$

use reduction f

then use alg above

Proof 2: A in NP

\exists poly time alg.

input x to problem A

\exists hint y , such that

in poly time y proves x is "yes"

Exponential time alg: try all possible y 's

c constant

n = input length

Methods that often work well in practice, even if not guaranteed

Practical algorithms for some NP-complete problem

1. SAT

2. Integer programming

ORIE 3300, 3310

today: use integer programming

given: variables $(x_1, \dots, x_n) = x$

$$Ax \leq b$$

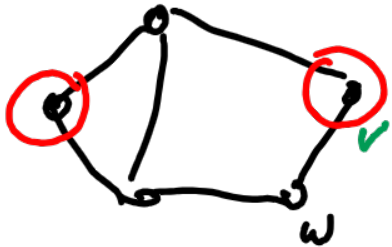
$$\max w^T x = \text{value}$$

x integer (often $x \in \{0, 1\}$)

note version allowing fractions poly time

Example: Independent set

Example:
independent set
 $G = (V, E)$



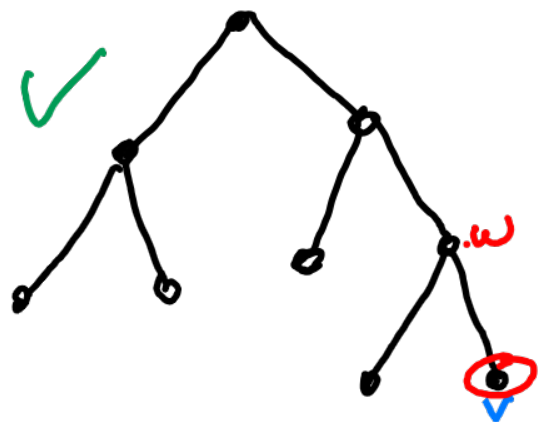
\equiv variables $x_v \in \{0, 1\}$
"1 if included 0 if not"
 $x_v + x_w \leq 1$ all $e = (v, w)$

$$\# \sum_v x_v = k$$

Claim: integer program
equivalent to indep set

Solvable special case?

Solve indep set graph $G=(V,E)$ is a tree



Question is Indep Set NP-complete if G is a tree?

No

Note: a tree has a degree \leq unode

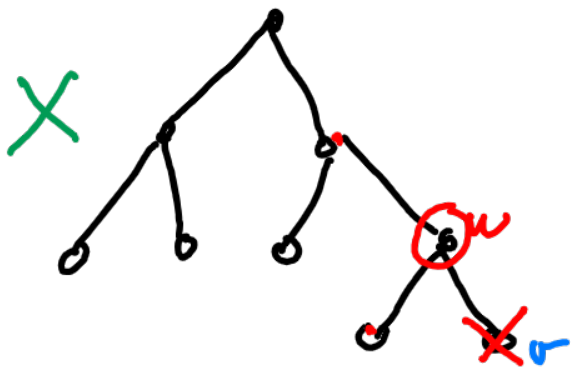
Should I include v in indep set?

if yes \Rightarrow parent of v : w not allowed

if not if w not included then excluding is waste

\rightarrow should include w

v excludes a subset of what w excludes (if including themselves in the set)
 \Rightarrow including v better than w



Solvable special case?

Algorithm:

Indep Set in tree G

select a leaf v or isolated node

include in indep set

delete parent

Indep Set in remaining graph

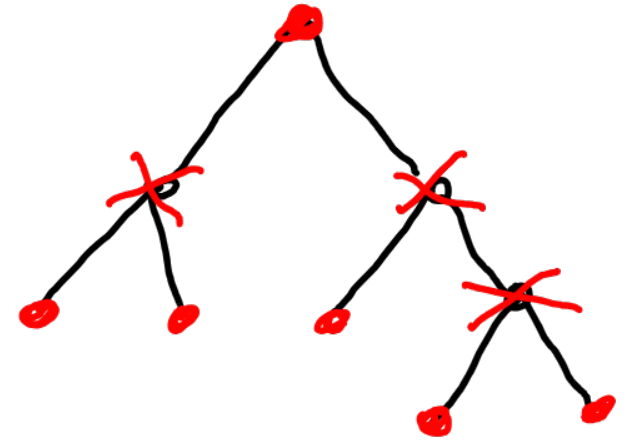
Algorithm finds maximum size independent set

Proof: by induction

Summary (today)

1. notice special structure & use

2. use practical SAT & integer programming solvers



Pre-enroll comments related to 4820

Fall

CS 4814 complexity
4820 TA

2850 Networks
6850 —

Spring

Michal Kim
Bobby Kleinberg
Info course
Jon Kleinberg

CS 4830 crypto

6820 grad algorithm