Greedy Algorithm

- Minimize $\sum C_i f_i$
- Sort in decreasing order of $\frac{f_i}{b_i}$.

- Consider any alternate ordering, ALT
- ALT must have some $i, j = i + 1$ with $f_j / b_j \geq f_i / b_i$
- Suppose we swap them to form ALT$'$.

- Consider any alternate ordering, ALT
- ALT must have some $i, j = i + 1$ with $f_j / b_j \geq f_i / b_i$
- Suppose we swap them to form ALT$'$.
- Let C be completion of pizza before i, j
- Total cost was $P + f_i(C + b_i) + f_j(C + b_i + b_j)$

- Consider any alternate ordering, ALT
- ALT must have some $i, j = i + 1$ with $f_j / b_j \geq f_i / b_i$
- Suppose we swap them to form ALT$'$.
- Let C be completion of pizza before i, j
- Total cost was $P + f_i(C + b_i) + f_j(C + b_i + b_j)$
- New cost is $P + f_i(C + b_i + b_j) + f_j(C + b_j)$
- New - Old $= f_i \cdot b_j - f_j \cdot b_i$
- Because $f_j / b_j \geq f_i / b_i$, difference is non-negative.

Divide and Conquer

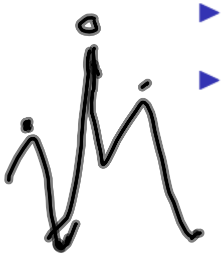
- Find a pair i and j, with $i < j$
- Maximize the value of $S_i - S_j$
-

10	3	27	20	13	8	14	11	25	36

Divide and Conquer

See correction on Piazza

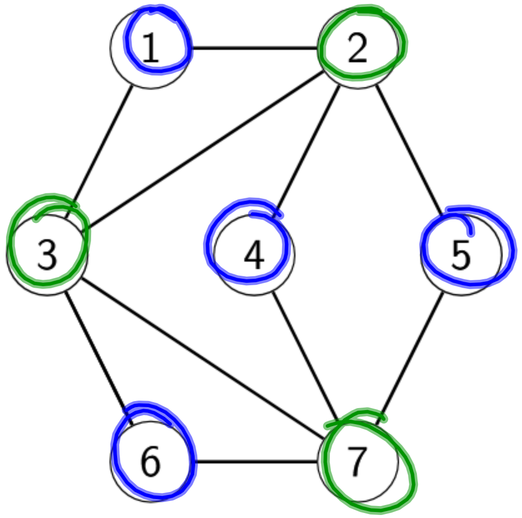
- If small enough, use brute force
- Let (l_1, h_1) be the solution in LHS
- Let (l_2, h_2) be the solution in RHS
- Return best pair of $(h_1, l_1), (h_2, l_2), (\cancel{h_1, l_2})$



- $T(n) = 2T(n/2) + O(1)$

Independent Set and Vertex Cover

Find an independent set of size 4 and a vertex cover of size 3 in this graph:



Can you confirm a solution?

- Suppose I claim G has an INDEPENDENT SET of size k. What evidence should I provide of my claim?
- Could you write an algorithm to verify such a claim?
  Input: G, k, and the evidence from the first point.
  Output: True or false, indiciating if the evidence really confirms an independent set of size k.

Verifier for INDEPENDENT SET

Certificate: V', a set of vertices.
Verifier:

if $V' \not\subseteq V$ **then**
  return false
if $|V'| \neq k$ **then**
  return false
for all edges $e = (u, v) \in E$ **do**
  if $u \in V'$ and $v \in V'$ **then**
    return false
return true

Which is easier?

Which do you think is easier to write an algorithm for?

$\textsc{Vertex Cover}(G, k)$ $\qquad$ $\textsc{Independent Set}(G, k)$

Which is easier?

Which do you think is easier to write an algorithm for?

VERTEX COVER(G, k)	INDEPENDENT SET(G, k)
return IND. SET$(G, n-k)$	return VER. COV$(G, n-k)$

Coincidence?

Claim: a graph G has an INDEPENDENT SET of size k if and only if it also has a VERTEX COVER of size $n - k$.
Part 1 of proof: If a graph G has an INDEPENDENT SET of size k, it also has a VERTEX COVER of size $n - k$.

Coincidence?

Claim: a graph G has an INDEPENDENT SET of size k if and only if it also has a VERTEX COVER of size $n - k$.
Part 2 of proof: If a graph G has VERTEX COVER of size $n - k$, it also has a an INDEPENDENT SET of size k.

Set Cover

Can you select three of the following sets in such a way that each letter from ‘A’ through ‘J’ is in at least one chosen set?

Set Number	Elements
1	A B
2	A C D E
3	B C F I
4	D G
5	E H
6	I J
7	F G H J

Is Set Cover easier or harder than Vertex Cover?

Can we use SET COVER to solve VERTEX COVER?

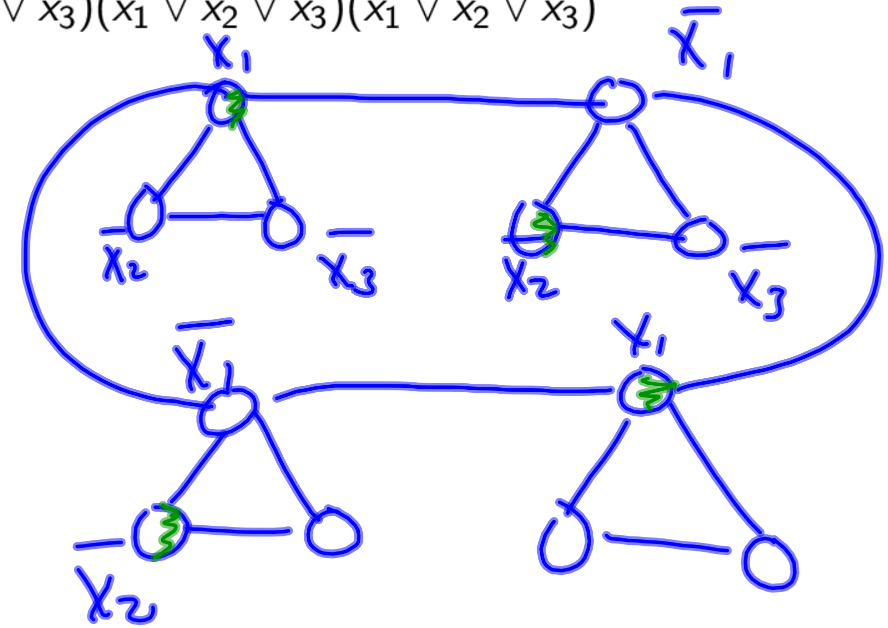
Imagine I give you a solution to SET COVER and solve:

VERTEX COVER(G, k)

3-Sat

$$\phi = (x_1 \vee \overline{x_2} \vee \overline{x_3})(\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})(\overline{x_1} \vee \overline{x_2} \vee x_3)(x_1 \vee x_2 \vee x_3)$$

Variable	Truth Value
x_1	True or False
x_2	True or False
x_3	True or False



Can Independent Set solve 3-Sat?

Imagine I give you a solution to INDEPENDENT SET. Let's use it to write an algorithm that solves 3-SAT.
Hint: Build a graph such that:

- The INDEPENDENT SET solver will select one variable from each clause, when given the graph and an appropriate value of k as input.
- Before you use the solver, modify the graph so that x_i and $\overline{x_i}$ won't both be selected, for any i.
- It is okay to select x_i two or more times, or $\overline{x_i}$ two or more times, as long as you don't select x_i and $\overline{x_3}$

Can Independent Set solve 3-Sat?

Given 3SAT instance
- Build G w/ 3K vertices (K = # clauses)
  - ea. clause is clique size 3
  - label w/ elts of that clause
- connect x_i — $\overline{x_i}$ for all i
- ind set (G, K)

Could we get false positives?

Show that, if that graph G has an independent set of size k, then the 3-SAT instance truly has a satifying assignment.

Could we get false negatives?

Show that, if the 3-SAT instance we have as input has a satisfying assignment, the graph we build will have an independent set of size k.

$$\begin{tikzcd} X \arrow[r] & y \end{tikzcd}$$

x solves y
poly + call to x

$$\begin{tikzcd} 3SAT & \text{Ind set} \arrow[l, bend right] \arrow[r, bend left] & VC \arrow[l, bend left] & \text{Set Cover} \arrow[l, bend right] \end{tikzcd}$$

What is a decision problem?

How might you expect to find INDEPENDENT SET as a problem?

- Does this graph G have an independent set of size k?

  decision

- Find the largest independent set in graph G.

  optimization

A stupid algorithm for Independent Set

for all subsets V' of V of size k **do**
  if V' is an Independent Set **then**
    return true
return false

- This takes time $O(n^k)$
- We could do similar for finding the largest independent set in time $O(2^n nk)$

What is $\mathcal{NP}$?

- $\mathcal{NP}$ stands for "Non-deterministic Polynomial"
- This is the set of decision problems *whose "yes" instances can be verified* in polynomial time.

How do we show a problem is in $\mathcal{NP}$?

- Certificate: the "proof" that this is a "yes" instance.
  - For example, V' in Independent Set
- Verifier: a polynomial time algorithm:
  - Input: the problem instance **and** the certificate
  - Output: true or false (if the certificate is valid)

Proof that Independent Set is in $\mathcal{NP}$

Certificate: V', a set of vertices.
Verifier:

if $V' \not\subseteq V$ **then**
  return false
if $|V'| \neq k$ **then**
  return false
for all edges $e = (u, v) \in E$ **do**
  if $u \in V'$ and $v \in V'$ **then**
    return false
return true

Prove that VERTEX COVER is in $\mathcal{NP}$

Certificate: V'

Verifier:

1. if $|V'| \neq k$, reject
2. If $V' \not\subseteq V$, reject
3. for each $e=(u,v) \in E$
   if $u \notin V'$ and $v \notin V'$
   reject
4. accept

Prove that SET COVER is in $\mathcal{NP}$

Set Cover: given $\{S_i\}$, K. Ea. $S_i \subseteq \mathcal{U}$

Certificate: The subsets chosen

Verifier:

1. If not really subsets from input, reject
2. If $\neq K$ ~~$> K$~~ given, reject
3. If $\bigcup_{\text{chosen}} S_i \neq \mathcal{U}$ reject
4. accept

Prove that 3-SAT is in $\mathcal{NP}$

Cert: Truth Value Assignment

Verifier:
1. If any var unassigned, reject
2. for ea. clause C
   if C unsatisfied
   reject
3. accept

The Cook-Levin Theorem

If $X \in NP$, then this:

X() { .... 3-SAT(); }

in poly time + call to 3-SAT.

Proof that Independent Set is $\mathcal{NP}$-complete

- IND SET ∈ NP (see earlier)

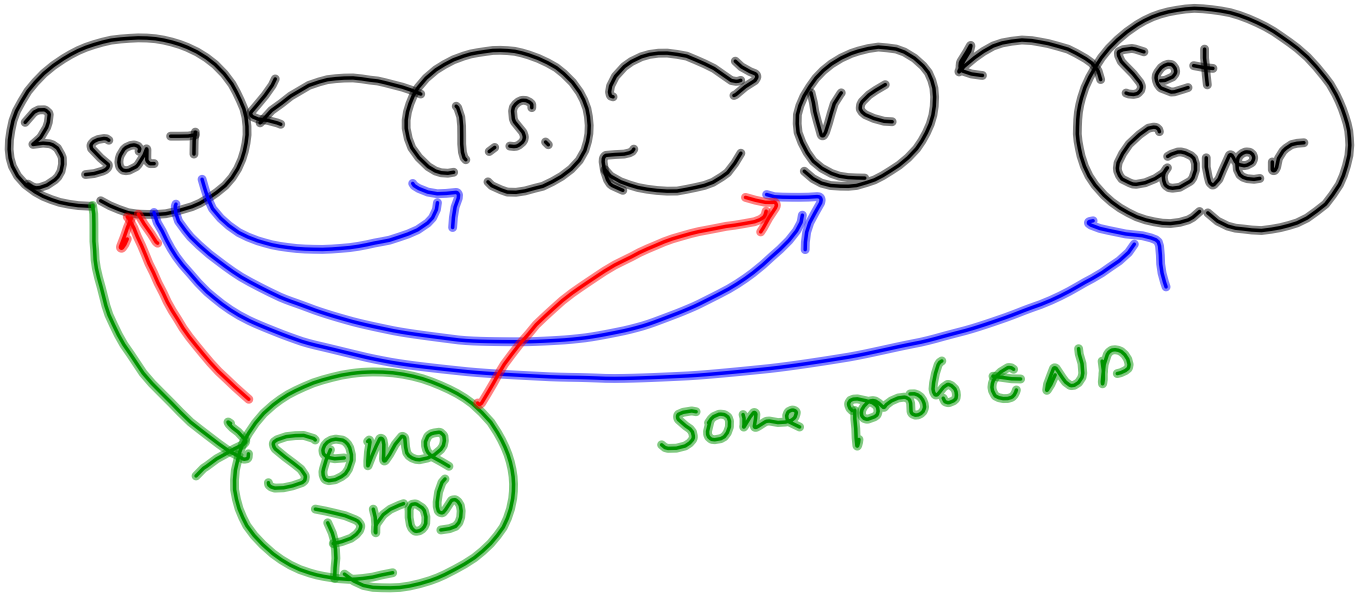
reduction
- 3SAT ( )
  ⋮
  IND SET

(see earlier)

- reduction is correct
  - no false positives
  - no false negatives

What have we seen so far?

Cook - Levin



What is the set $\mathcal{P}$?

$\mathcal{P}$: polynomial time solvable.

$P \stackrel{?}{=} NP$