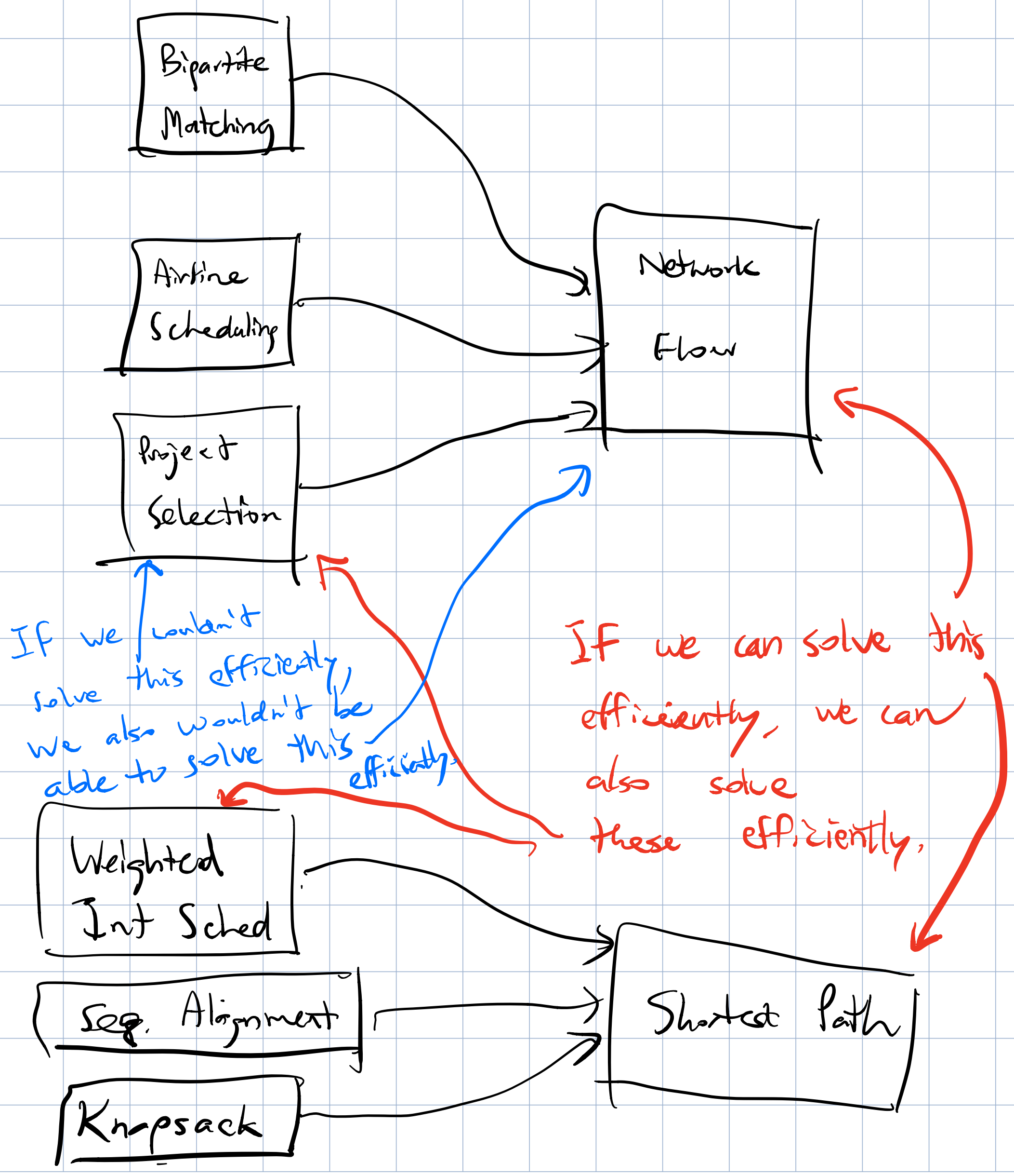
29 Mar 2021 NP-Completeness (§8.1)

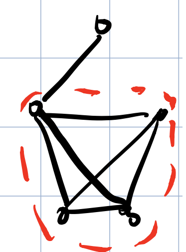
Reductions let us re-use one algorithm (e.g. max-flow) to solve other computational problems (e.g. airline scheduling).



Some problems currently believed to be computationally hard.

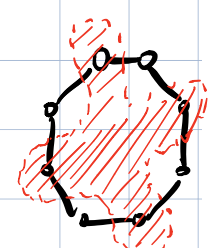
Given undirected graph G and a natural number k...

CLIQUE: does G has a set of k vertices that are all joined to one another by edges? (each pair of vertices in the set are joined by an edge)



a clique of size 4

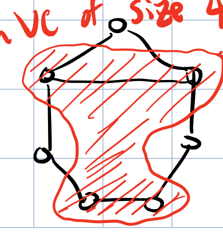
INDEPENDENT SET: does G has a set of k vertices that have <u>no</u> edges between them?



an indep't set of size 4.

VERTEX COVER: does G have a set of k vertices that <u>covers</u> every edge? (each edge has ≥ 1 endpoint in the set)

a VC of size 4



These are all "computationally equivalent": an efficient algorithm for one would give us a way to solve all the others efficiently as well.

Let $\overline{E} = \{ (u,v) \mid (u,v) \text{ is not in } E \}$

Observe: $G = (V,E)$ has a clique of size k
$\iff \overline{G} = (V, \overline{E})$ has independent set of size k

Reduction

$\boxed{\text{CLIQUE}} \longleftrightarrow \boxed{\text{INDEP SET}}$

$(G,k) \longmapsto (\overline{G}, k)$

$(\overline{G}, k) \longleftarrow\!\dashv (G,k)$

$\overline{G}$: G with the complementary edge set, $\overline{E}$.

<u>Observe:</u> S is an indep set in $G = (V,E)$
$\iff V - S$ is a vertex cover in G

<u>Proof.</u> Suppose S is indep set and $e = (u,v)$ is an edge. The endpts of e can't both belong to S so one of them belongs to $V - S$

$\therefore$ V-S is a vertex cover.

Conversely if V-S is a vertex cover and (u,v) is an edge, then at least one of u,v belongs to V-S so they don't both belong to S.

$\therefore$ S is an indep set.

<u>Reduction</u>.

$$\boxed{\text{VERTEX COVER}} \longleftrightarrow \boxed{\text{INDEP SET}}$$

$$(G, k) \longmapsto (G, n-k)$$

where $n = \#$ vertices.

$$(G, n-k) \longleftarrow\!\!\!\!\mid (G, k)$$

What do we know about solving these?

Brute force: $O\left(m \cdot \binom{n}{k}\right) = O(m \cdot n^k)$.

Fastest known: $O(n^{c \cdot k})$ for some $0 < c < 1$.

A poly-time algorithm would need a <u>constant</u> in the exponent, not a multiple of k.

3 SAT. Given n Boolean variables $x_1, x_2, \dots, x_n$ and their negations denoted $\overline{x_1}, \overline{x_2}, \dots, \overline{x_n}$ (Collectively these $2n$ terms are called "literals")

Given m "clauses" each formed by connecting ≤ 3 literals using Boolean OR operation

e.g. $x_4 \vee \overline{x_9} \vee \overline{x_{10}}$

Does $\exists$ a truth assignment of $x_1, \dots, x_n$ that satisfies every clause?

Ex. $(x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_4) \wedge (\overline{x_3} \vee \overline{x_4})$

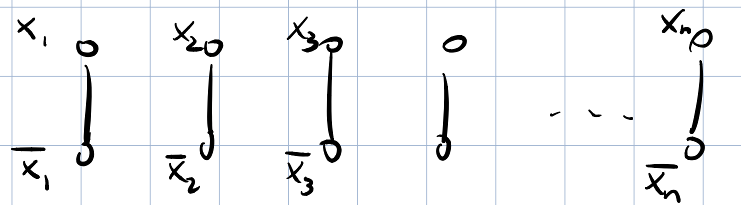
is satisfied by $(x_1, x_2, x_3, x_4) = (T, F, T, F)$

$(x_1 \vee x_2) \wedge (\overline{x_2} \vee x_3) \wedge (\overline{x_3} \vee \overline{x_1})$ is satisfied by $(x_1, x_2, x_3) = (T, F, F)$.

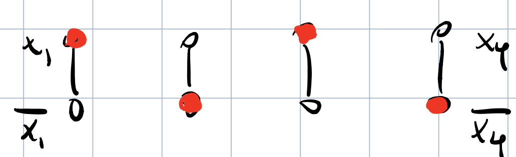
$\boxed{\text{3 SAT}} \longrightarrow \boxed{\text{independent set}}$

Assume you had an algorithm to solve independent set. Try using it to solve 3SAT.

(1) This subroutine that is good at finding large vertex sets ... how to manipulate it into finding a truth assignment of n variables?



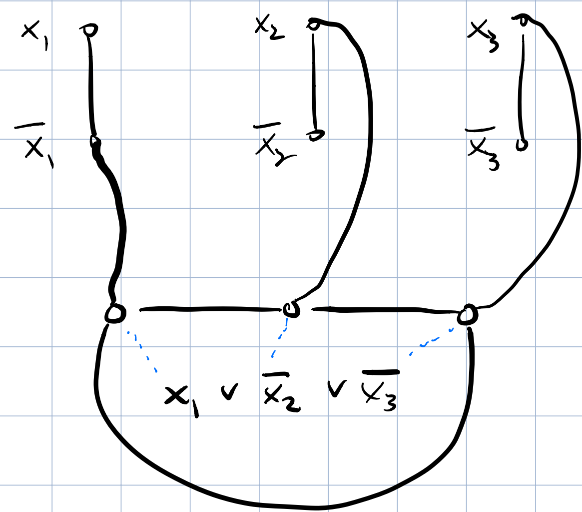
choosing an indep set of size n in this graph is equivalent to choosing a truth assignment of $x_1, \ldots, x_n$.



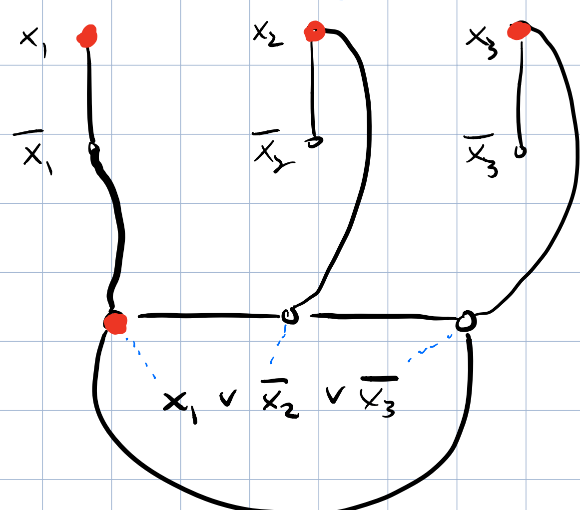
corresponds to $(x_1, x_2, x_3, x_4) = (T, F, T, F)$.

(2) How to connect the "variable assignment gadgets" with connective tissue such that only the truth assignments that satisfy

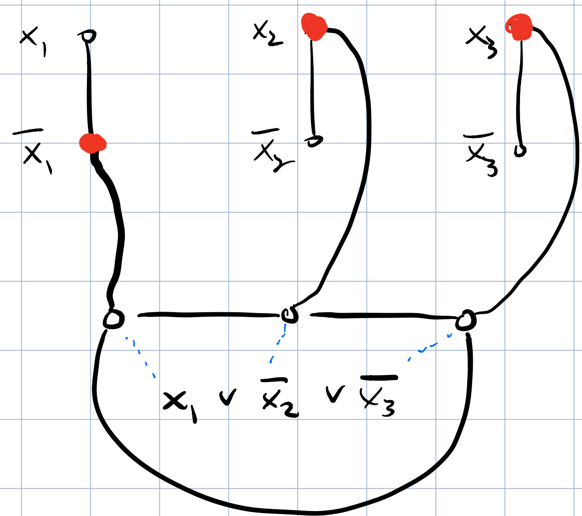
every clause lead to large independent sets?



A truth assignment that satisfies the clause corresponds to a vertex set that lets me choose one of the three "hull vertices"



A truth assignment that violates the clause prevents choosing any "hull vertex".



So transform 3SAT formula with n vars, m clauses to INDEP SET problem $(G, n+m)$ where G is as illustrated above.