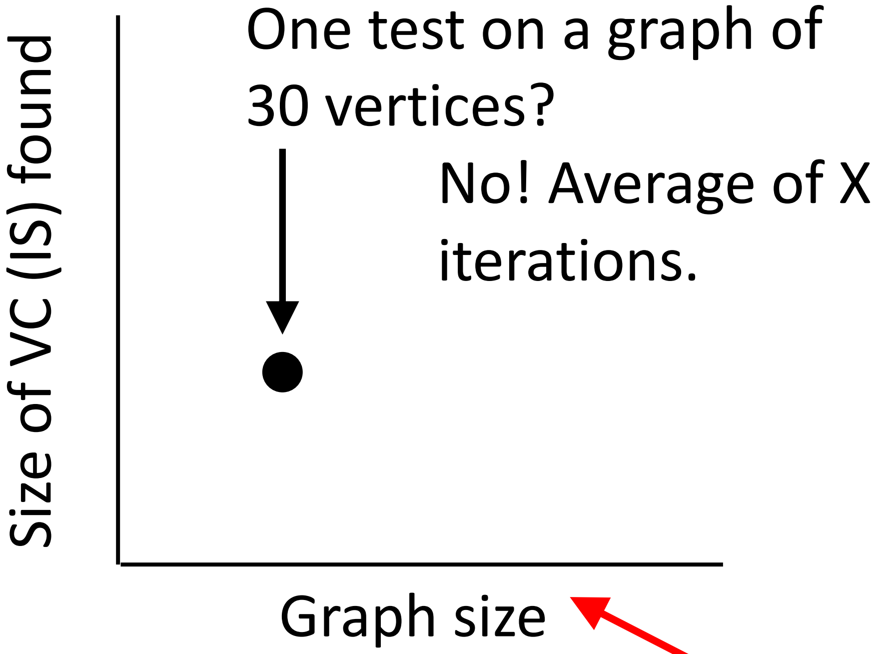
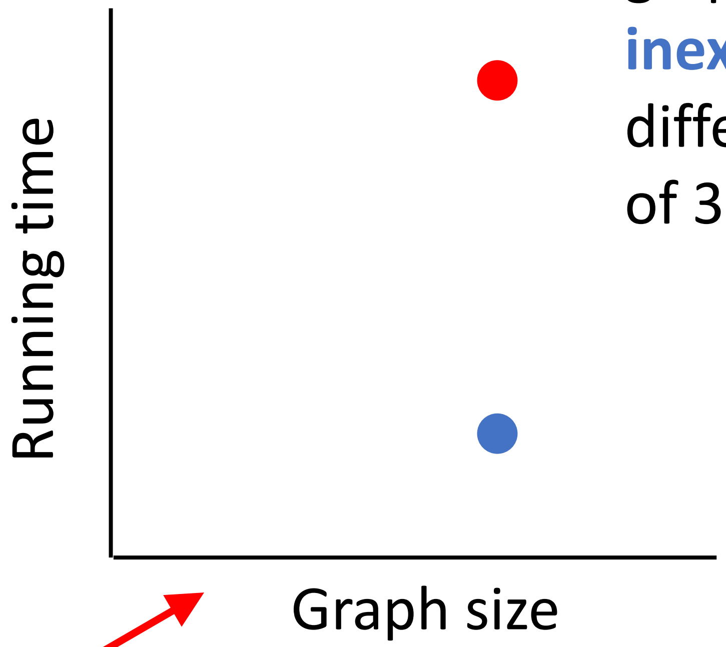
CLIQUE
CSCI 338

Project 3

What performance metrics do we care about?

Accuracy, speed.





Exact result on random graph of 30 vertices and **inexact result** on different random graph of 30 vertices?

No! results on same random graph.

"Size" of graph (# vertices, # edges, connectivity)

NP-Complete

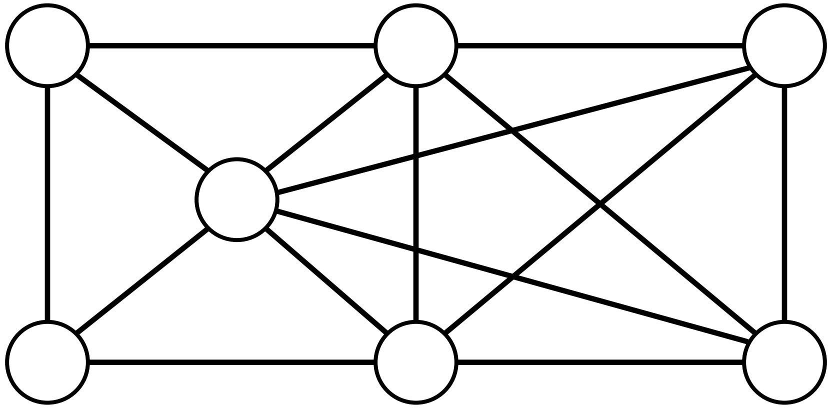
How to show problem B is in NP-Complete:

1. Show B is in NP.
2. Pick some known NP-Complete problem A.
3. Show how generic instances of A can be translated into instances of B.
4. Show that the translation process runs in polynomial time.
5. Show that if the answer to A’s instance is ‘yes’, the answer to B’s instance is also ‘yes’.
6. Show that if the answer to B’s instance is ‘yes’, the answer to A’s instance is also ‘yes’.

$CLIQUE$

Clique: a subgraph where every pair of nodes share an edge (i.e. a complete subgraph).

k-Clique: A clique that contains k vertices.



$CLIQUE = \{\langle G, k \rangle : G$ is an undirected graph with a k−clique$\}$

$CLIQUE$

Claim: $CLIQUE \in NP$-Complete

Proof:

$CLIQUE$

Claim: $CLIQUE \in NP$-Complete

Proof:

1. $CLIQUE \in NP$

$CLIQUE$

Claim: $CLIQUE \in NP$-Complete

Proof:

1. $CLIQUE \in NP$

   Given a graph $G = (V, E)$, where $|V| = n$, and a subset $S \subseteq V$, where $|S| \geq k$, check if all pairs of vertices in S are in E. Running time: $O(n^2)$.

$CLIQUE$

Claim: $CLIQUE \in NP$-Complete

Proof:

1. $CLIQUE \in NP$

   Given a graph $G = (V, E)$, where $|V| = n$, and a subset $S \subseteq V$, where $|S| \geq k$, check if all pairs of vertices in S are in E. Running time: $O(n^2)$.

2. ???

$CLIQUE$

Claim: $CLIQUE \in NP$-Complete

Proof:

1. $CLIQUE \in NP$

   Given a graph $G = (V, E)$, where $|V| = n$, and a subset $S \subseteq V$, where $|S| \geq k$, check if all pairs of vertices in S are in E. Running time: $O(n^2)$.

2. $3SAT \leq_P CLIQUE$

NP-Complete

How to show problem B is in NP-Complete:

1. ~~Show B is in NP.~~
2. Pick some known NP-Complete problem A.
3. Show how generic instances of A can be translated into instances of B.
4. Show that the translation process runs in polynomial time.
5. Show that if the answer to A's instance is ‘yes’, the answer to B's instance is also ‘yes’.
6. Show that if the answer to B's instance is ‘yes’, the answer to A's instance is also ‘yes’.

$CLIQUE$

Claim: $3SAT \leq_P CLIQUE$

Proof:

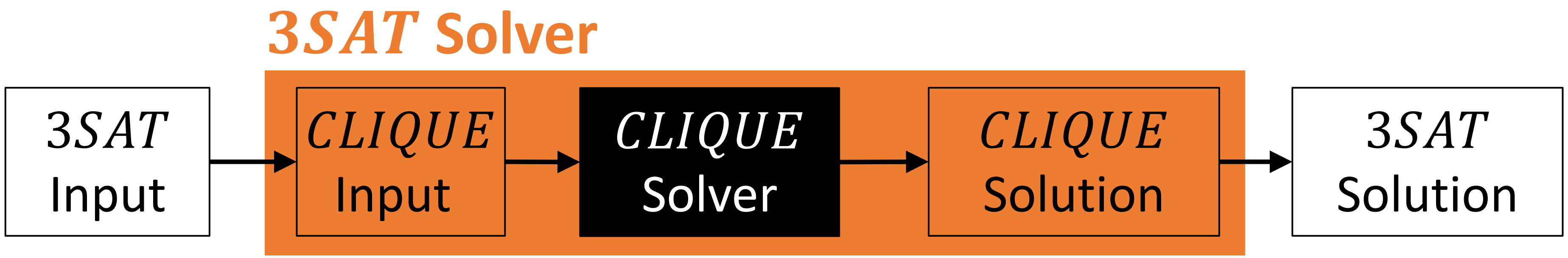
How to show problem B is in NP-Complete:

1. ~~Show B is in NP.~~
2. ~~Pick some known NP-Complete problem A.~~
3. Show how generic instances of A can be translated into instances of B.
4. Show that the translation process runs in polynomial time.
5. Show that if the answer to A's instance is 'yes', the answer to B's instance is also 'yes'.
6. Show that if the answer to B's instance is 'yes', the answer to A's instance is also 'yes'.

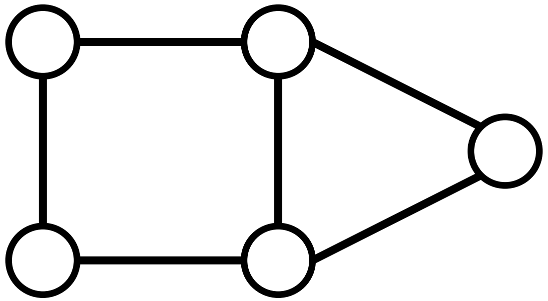
$CLIQUE$

Claim: $3SAT \leq_P CLIQUE$

Proof:



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



ϕ - Satisfiable $\iff$ $\exists\ k$ - Clique

$CLIQUE$

Claim: $3SAT \leq_P CLIQUE$

Proof: Let ϕ be a formula with k clauses. Generate an undirected graph G:

???

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

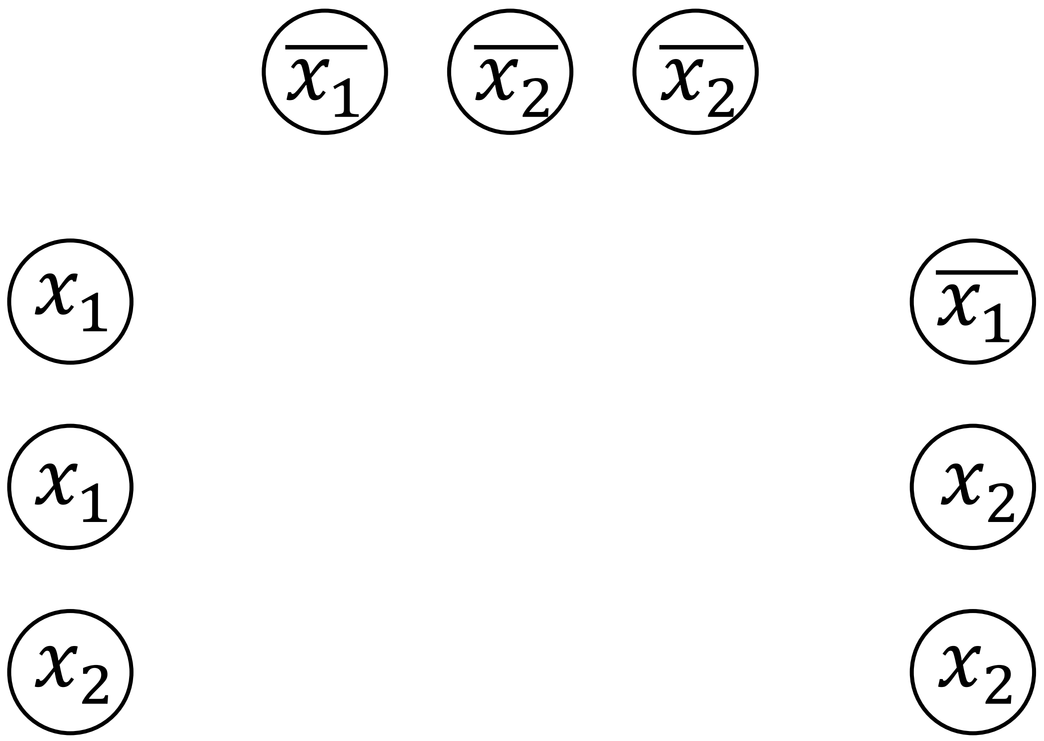
$CLIQUE$

Claim: $3SAT \leq_P CLIQUE$

Proof: Let ϕ be a formula with k clauses. Generate an undirected graph G:

For each clause in ϕ, make a node for each literal.

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



$CLIQUE$

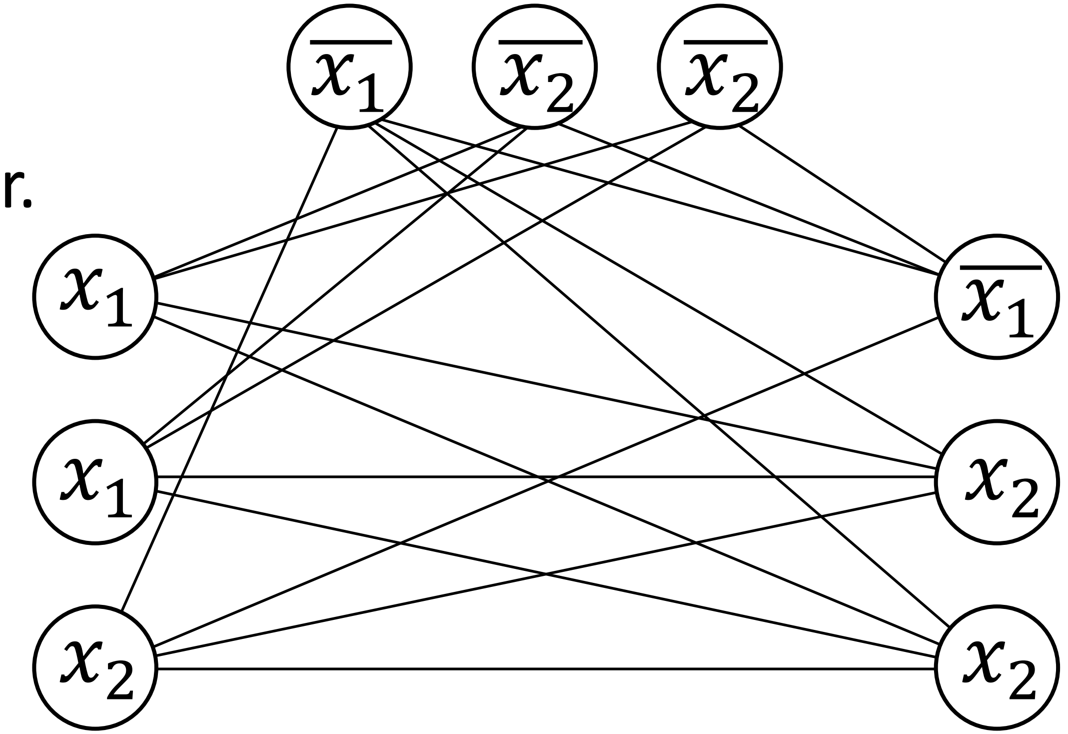
Claim: $3SAT \leq_P CLIQUE$

Proof: Let ϕ be a formula with k clauses. Generate an undirected graph G:

For each clause in ϕ, make a node for each literal. Make edge between every pair of nodes, except:

1. Nodes in the same clause
2. Nodes that are negations of each other.

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



NP-Complete

How to show problem B is in NP-Complete:

1. ~~Show B is in NP.~~
2. ~~Pick some known NP-Complete problem A.~~
3. ~~Show how generic instances of A can be translated into instances of B.~~
4. Show that the translation process runs in polynomial time.
5. Show that if the answer to A's instance is 'yes', the answer to B's instance is also 'yes'.
6. Show that if the answer to B's instance is 'yes', the answer to A's instance is also 'yes'.

$CLIQUE$

Claim: $3SAT \leq_P CLIQUE$

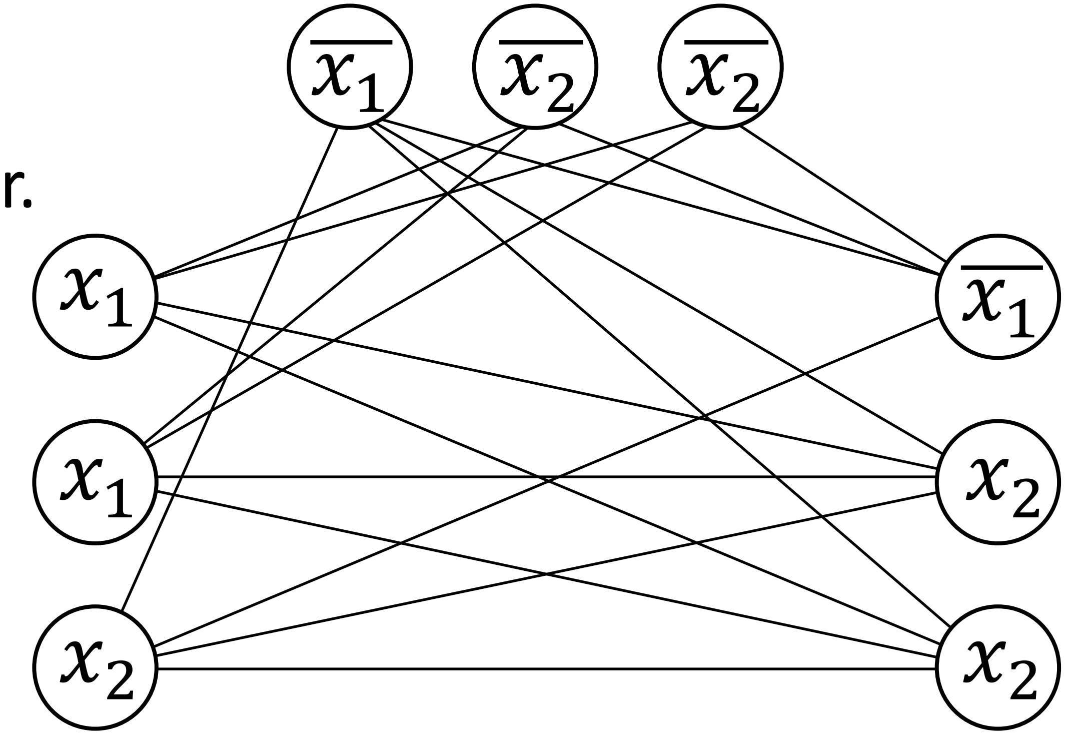
Proof: Let ϕ be a formula with k clauses. Generate an undirected graph G:

For each clause in ϕ, make a node for each literal. Make edge between every pair of nodes, except:

1. Nodes in the same clause
2. Nodes that are negations of each other.

Polynomial Time?

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



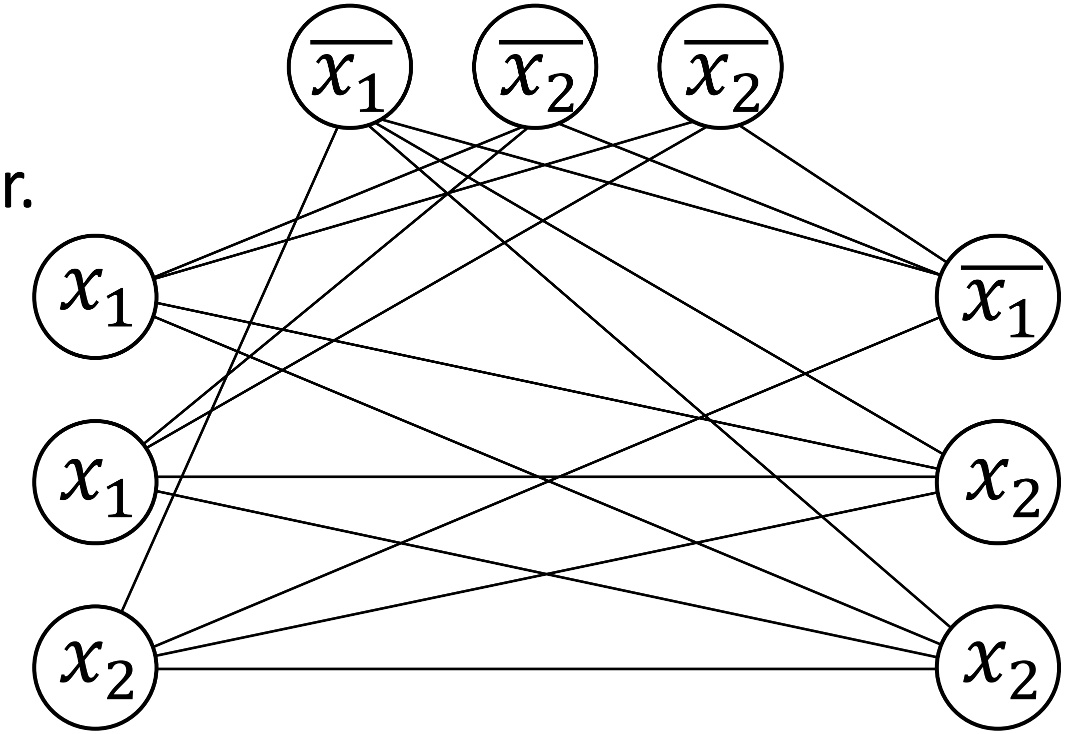
$CLIQUE$

Claim: $3SAT \leq_P CLIQUE$

Proof: Let ϕ be a formula with k clauses. Generate an undirected graph G:

For each clause in ϕ, make a node for each literal. Make edge between every pair of nodes, except:

1. Nodes in the same clause
2. Nodes that are negations of each other.



Polynomial Time?

Yes. (k = num clauses)
- **$3k$ nodes**
- **$O(k^2)$ edges (fewer than complete graph)**

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

NP-Complete

How to show problem B is in NP-Complete:

1. ~~Show B is in NP.~~
2. ~~Pick some known NP-Complete problem A.~~
3. ~~Show how generic instances of A can be translated into instances of B.~~
4. ~~Show that the translation process runs in polynomial time.~~
5. Show that if the answer to A's instance is 'yes', the answer to B's instance is also 'yes'.
6. Show that if the answer to B's instance is 'yes', the answer to A's instance is also 'yes'.

$CLIQUE$

Claim: $3SAT \leq_P CLIQUE$

Proof: Let ϕ be a formula with k clauses. Generate an undirected graph G:

For each clause in ϕ, make a node for each literal. Make edge between every pair of nodes, except:

1. Nodes in the same clause
2. Nodes that are negations of each other.

Need to show: $\boldsymbol{\phi}$ is satisfiable $\Leftrightarrow$ $\boldsymbol{G}$ has a $\boldsymbol{k}$-clique.

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