May 3rd, 2016

Exam 3. Thursday May 5th <u>12:20 – 1:50 pm</u>

Double sided handwritten cheating sheet

Calculator

5 problems

1 Shortest path

2 Random Sampling

3 Maxflow

4. String Matching Algorithm

5. Surprise

P: deterministic polynomial time solvable

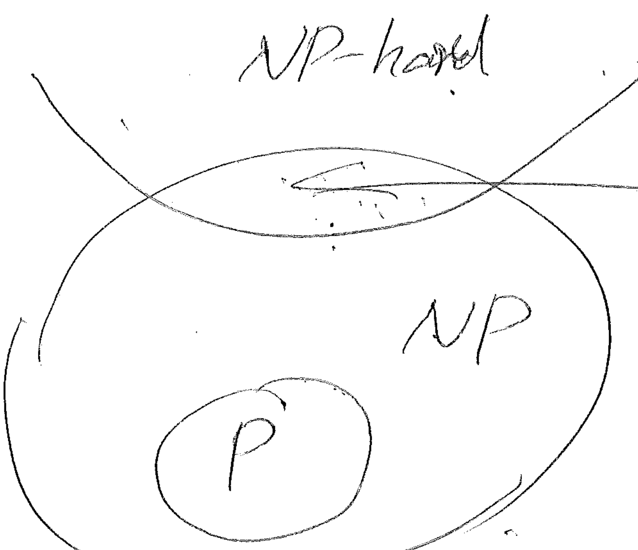
any problem that can be solved in polynomial time deterministically.

NP: Non-deterministic Polynomial Time Solvable

Polynomial Time verifiable

NP-hard: a problem is NP-hard if we can reduce every NP problem to it

A is NP hard, then for $\forall B \in NP$, $B \leq_p A$



NP-Complete: the hardest problems in NP

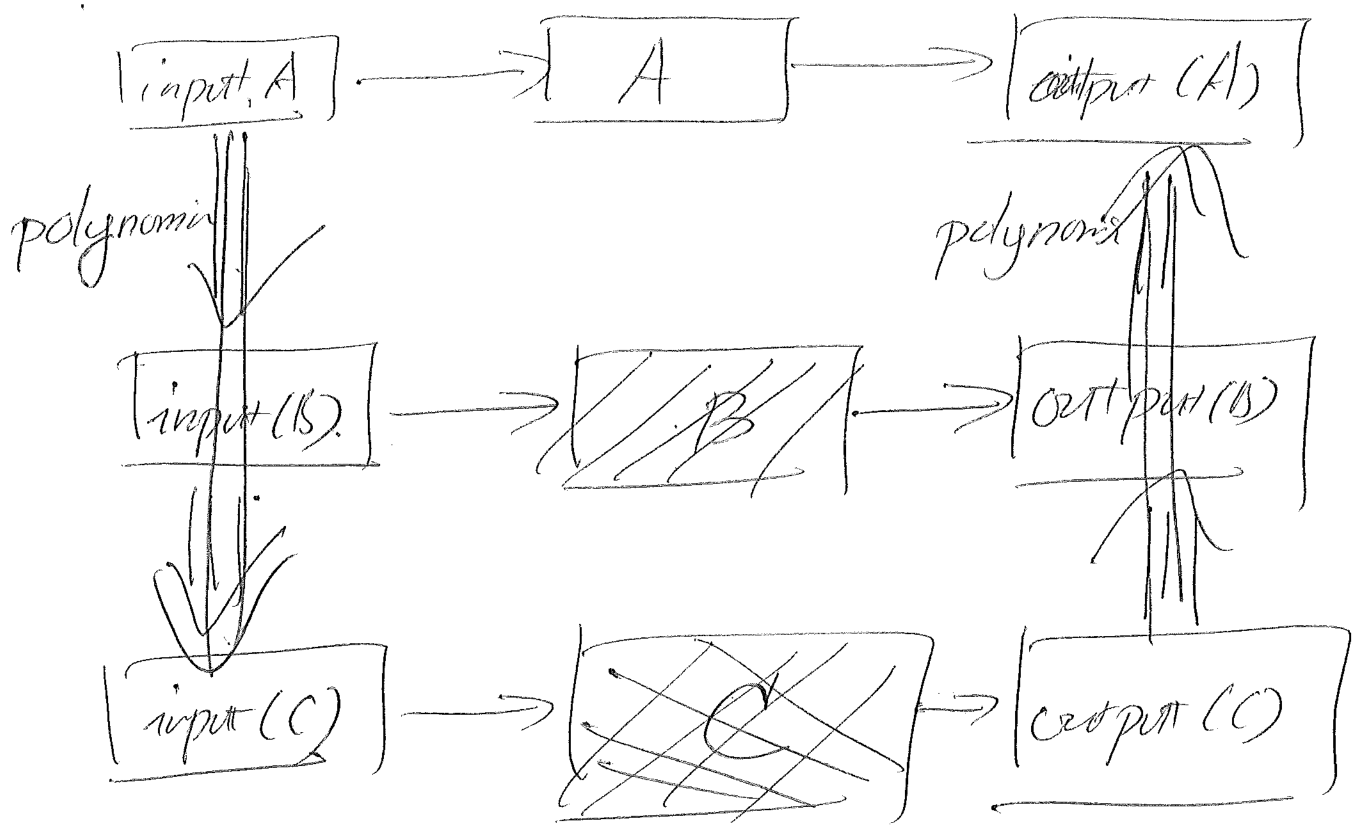
Conjecture: NP-C problems can't be solved in polynomial time

How to prove a problem B is NP-C ?

① Need to show $B \in NP$ ✓

② $B \in NP\text{-hard}$

$A \le_p B$ $\quad B \le_p C \Rightarrow A \le_p C$



$\le_p$ is transitive

Suppose we know that problem A is NP-hard.
This means every problem in NP can be reduced to A. , and $A \leq_p B$

$$\forall C \ (C \in NP) \rightarrow (C \leq_p A)$$

$$A \leq_p B$$

$\therefore \ \forall C \ (C \in NP) \rightarrow (C \leq_p B).$

$\therefore$ B is NP-hard

SAT

Truth Assignment

Let $X = \{x_1, \cdots, x_n\}$ be a set of n Boolean variables

A truth assignment of X is an assignment of 0 or 1 to each x_j.

A Clause C of length l is the disjunction (or) of l terms $t_1 \vee t_2 \vee \cdots \vee t_l$ where each

$t_i \in \{x_1, x_2, \cdots, x_n, \overline{x_1}, \overline{x_2}, \cdots \overline{x_n}\}$

$\overline{x_1} = \neg x_1$, $\overline{x_2} = \neg x_2$

E.g. $x_1 \vee x_2 \vee \overline{x_3}$ is a clause of length 3

$x_1 \vee \overline{x_1} \vee x_2$

A clause C is satisfied by a truth assignment f, if C evaluates to $\underline{1}$ under rules of Boolean Logic.

$X = \{x_1, x_2, x_3\}$

f:

$x_1 = 1 \qquad \overline{x_1} = 0$

$x_2 = 0 \qquad \overline{x_2} = 1$

$x_3 = 1 \qquad \overline{x_3} = 0$

C: $x_1 \vee x_2 \vee \overline{x_3}$ ✓

$\overline{x_1} \vee \overline{x_2} \vee x_3$ ✓

SAT:

Given a set of clauses $C_1, C_2, \cdots, C_k$ over a set $X = \{x_1, \cdots, x_n\}$, is there a truth assignment that satisfies all k clauses?

Recall the independent set problem

Given an undirected $G(V,E)$

an independent set is a subset $S \subseteq V$, such no ~~ver~~ two vertices of S are adjacent

The decision independent set problem:

Given $G(V,E)$, $0 \le k \le |V|$, is there an independent set of size k?

Prove the decision independent set is NP-C

Pf: (1) Decision independent $\overset{?}{\in}$ NP ✓

(2) Decision independent set $\overset{?}{\in}$ NP-hard

$\forall C ((C \in NP) \Rightarrow (C \le_p \text{Decision Independent Set}))$

$$\left.\begin{array}{l} \forall C ((C \in NP) \Rightarrow (C \le_p SAT)) \\ (SAT \le_p \text{Decision Independent set}) \end{array}\right\} \Rightarrow$$

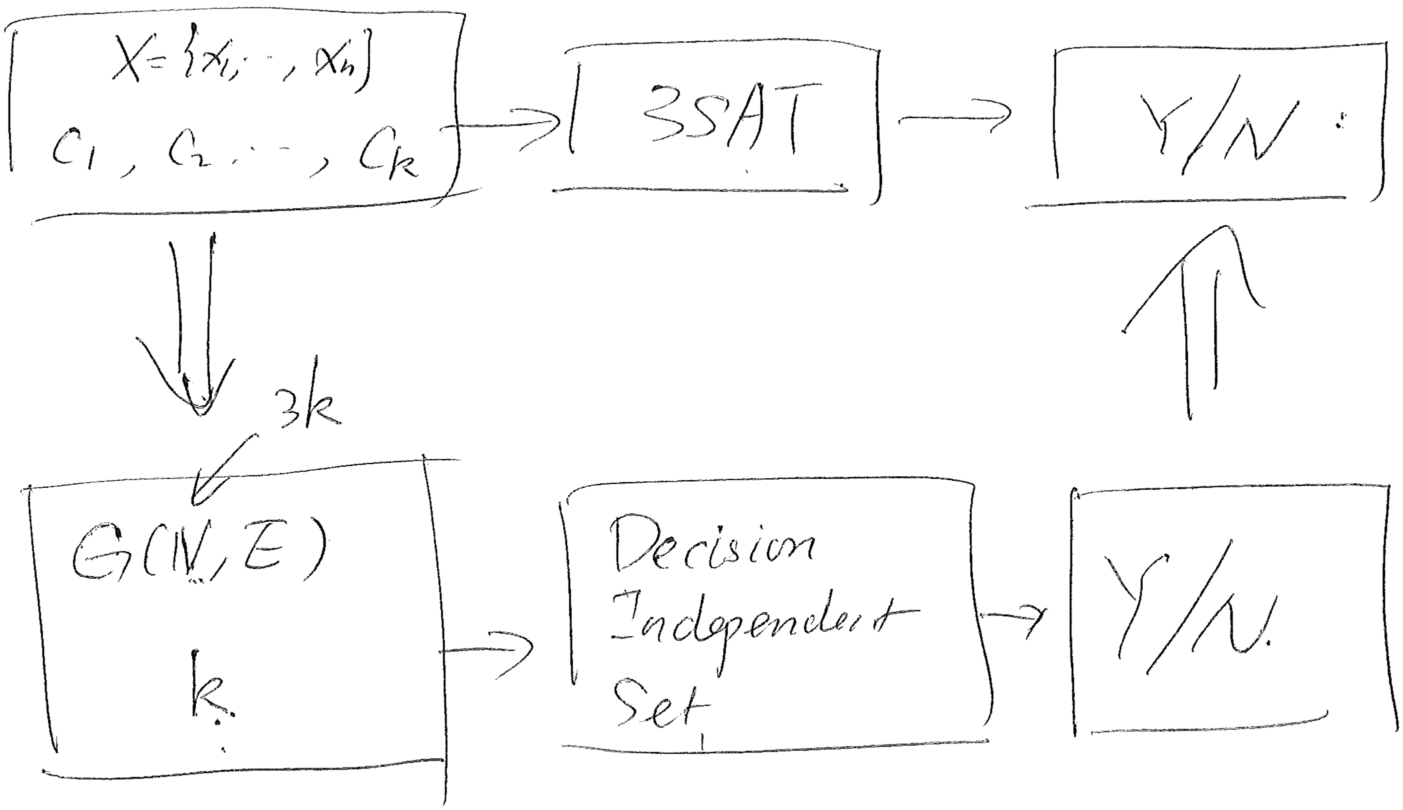
Strategy $SAT \le_p$ Decision Independent set

Need to reduce SAT to independent set.

in fact we will reduce 3-SAT to independent set

3-SAT: Given a set of k clauses each of length 3 over $X=\{x_1,\cdots,x_n\}$. Is there a truth assignment that satisfies all clauses

3SAT $\le_p$ independent set



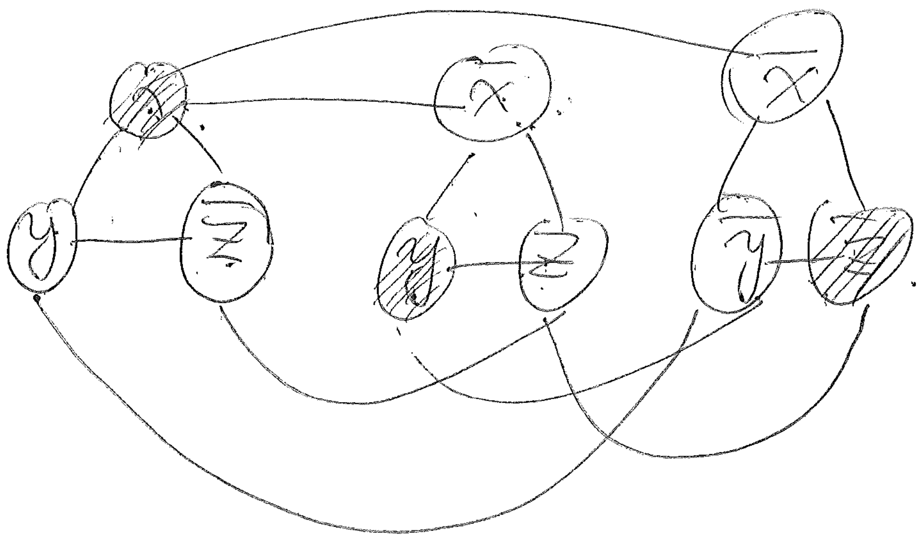
$x \vee y \vee \bar{z}$

$\bar{x} \vee y \vee z$

$\bar{x} \vee \bar{y} \vee \bar{z}$

$X = \{x, y, z\}$

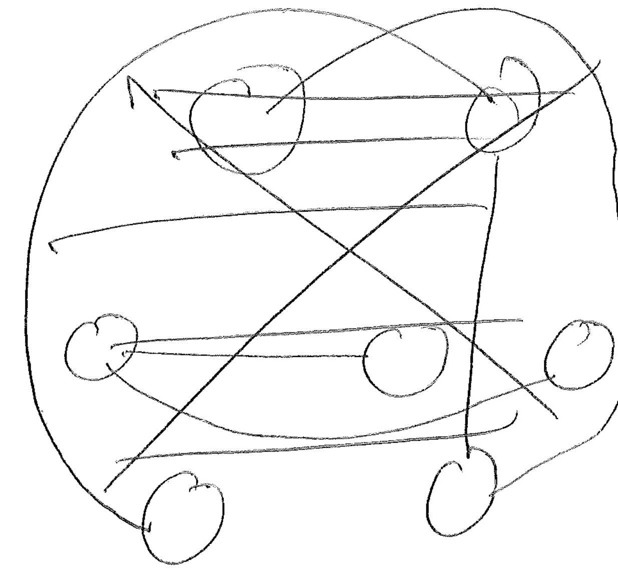
$\longrightarrow$ 3SAT $\longrightarrow$ Y/N

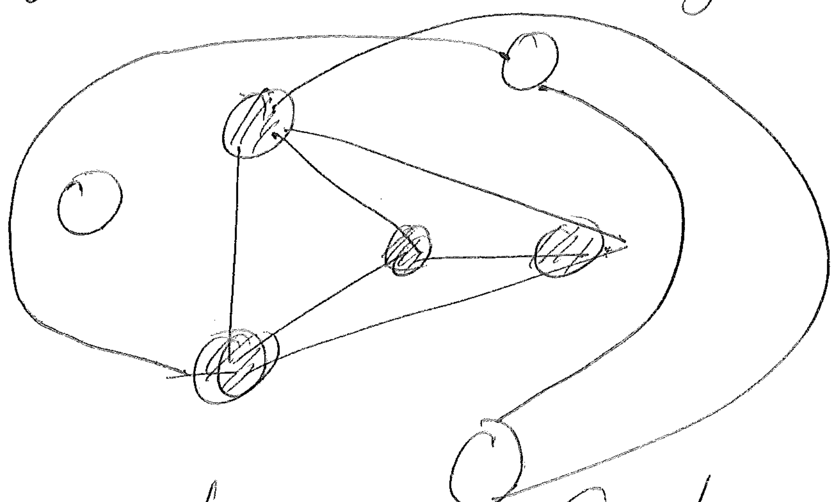


what does an independent set of size 3 mean?

The Clique Problem.

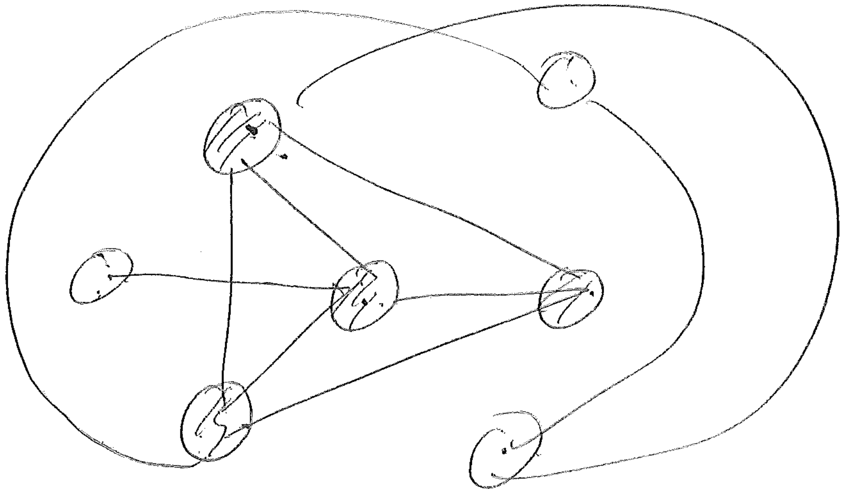
Given an undirected G(V,E)
a clique S is a subset of
vertices such that every pair
of vertices in S are adjacent



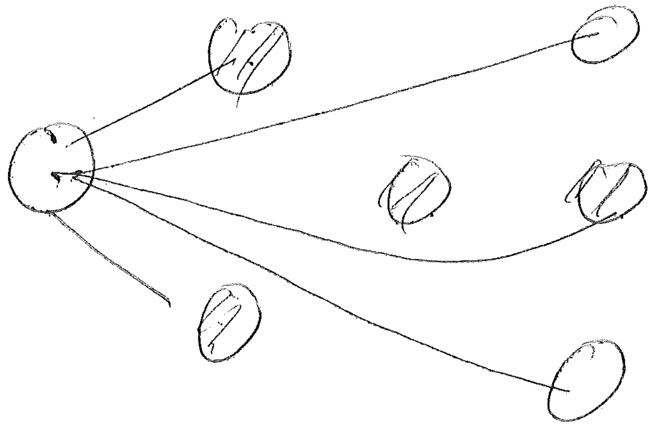


a clique is a Complete subgraph

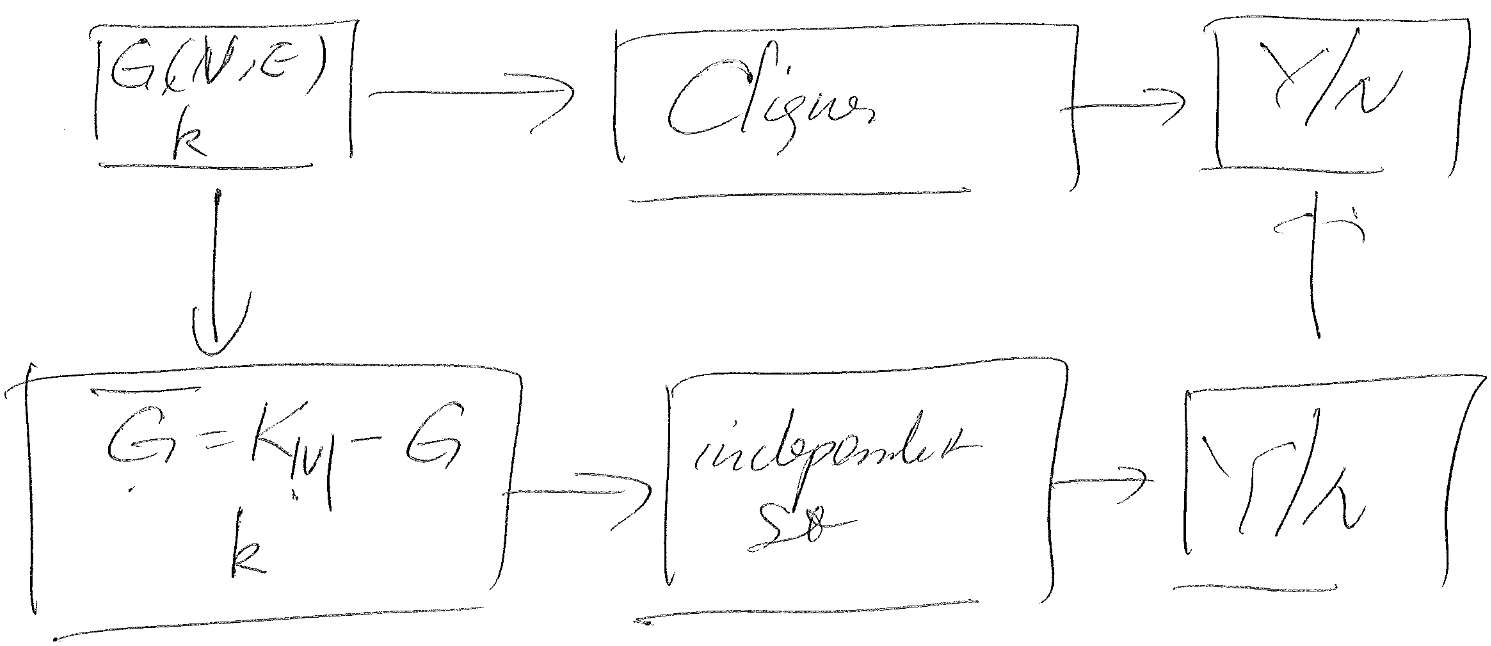
Decision version:
Give G(V,E), $0 \le k \le |V|$, is there a
clique of size k?
Is the problem in NP? ✓
Is the problem is NP-hard?
Can we reduce independent set to Clique?



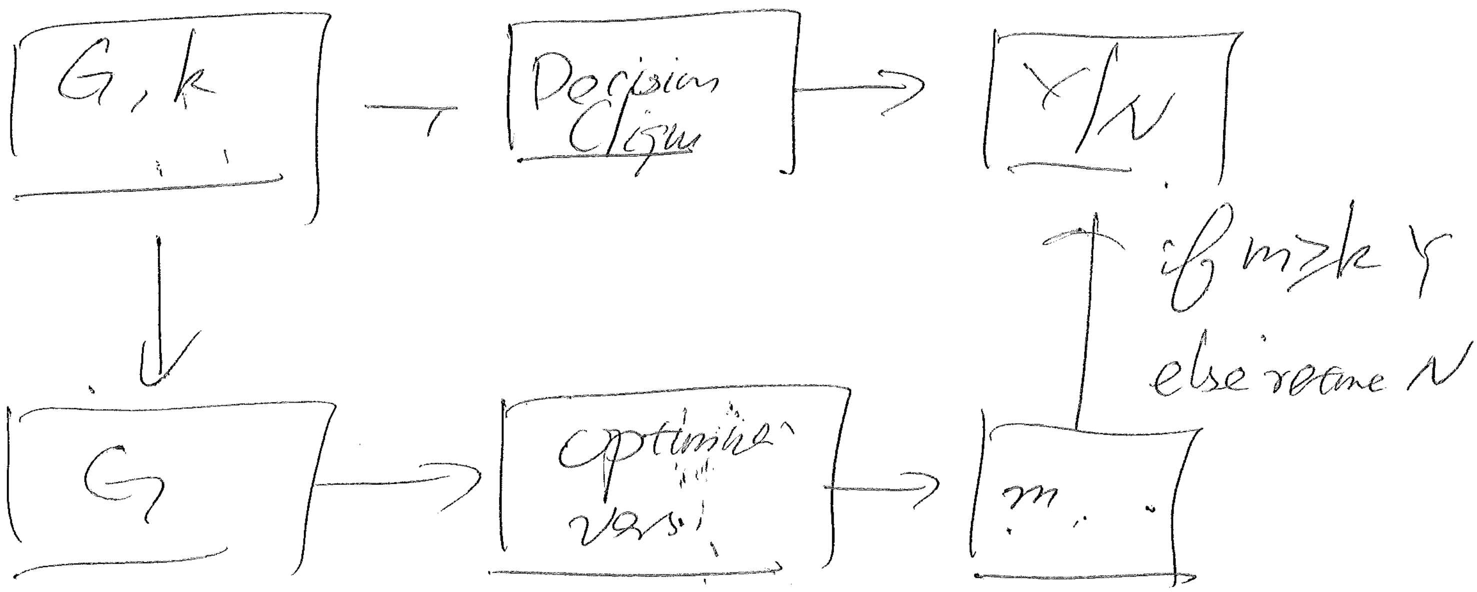
G



$\overline{G} = K_7 - G$



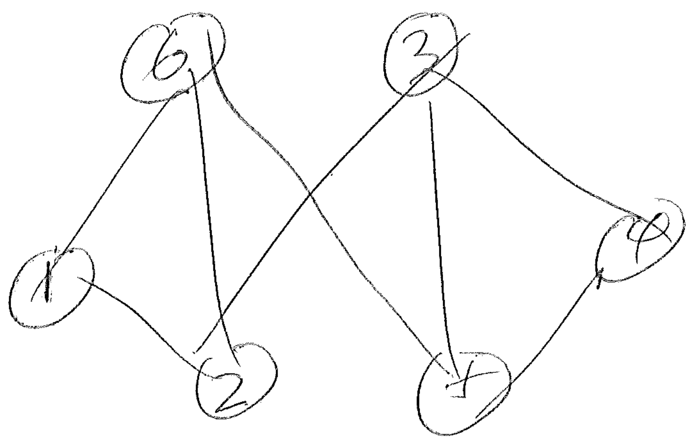
Optimization: given $G(V,E)$, what's the maximum size of the cliques in G?



Recall the Hamiltonian.

Give an undirected, unweighted Graph G(V,E)

Is there a simple cycle that goes through each vertex exactly once?



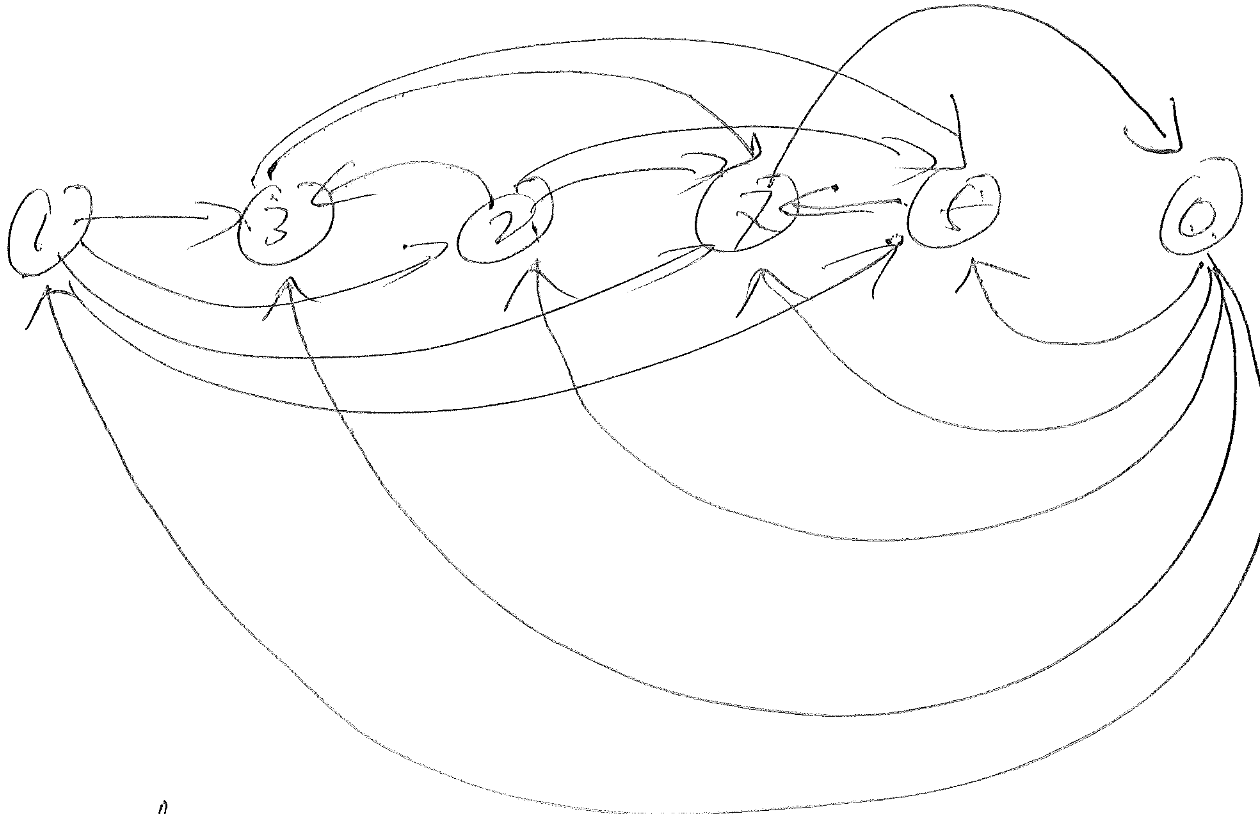
Hamiltonian is NP-C.

Consider the sorting problem: sort set distinct number

~~input a set of n distinct integers~~

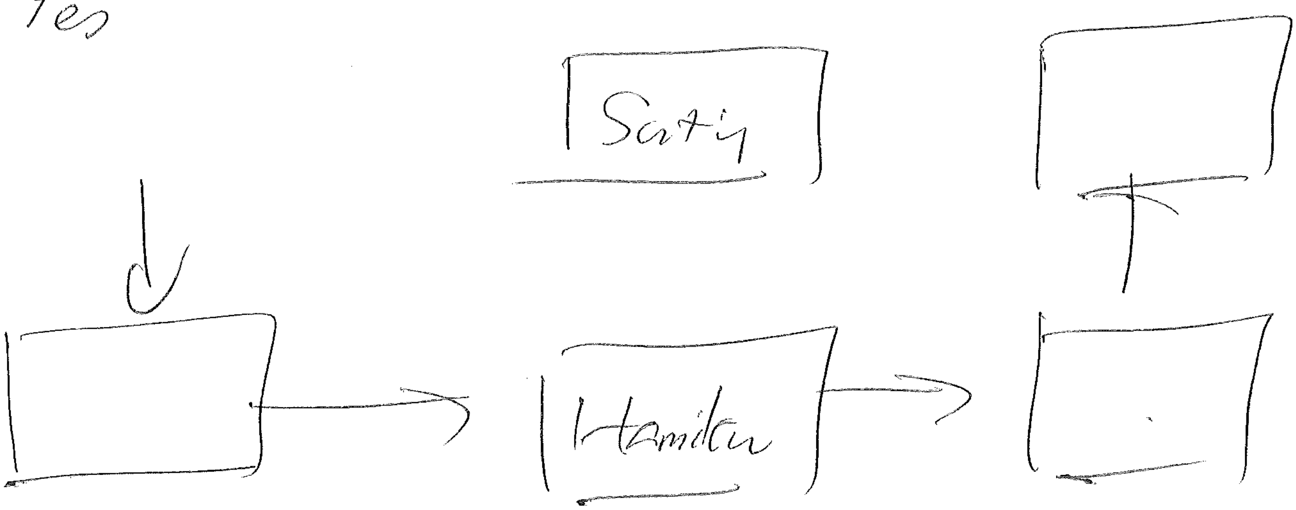
~~a permutation of {1, 2, ..., n}~~ output sorted

~~out the permutation~~



Does this graph has a directed Hamiltonian cycle?

Yes



sorting $\le_p$ Hamiltonian

Note this doesn't imply sorting is NP-hard.