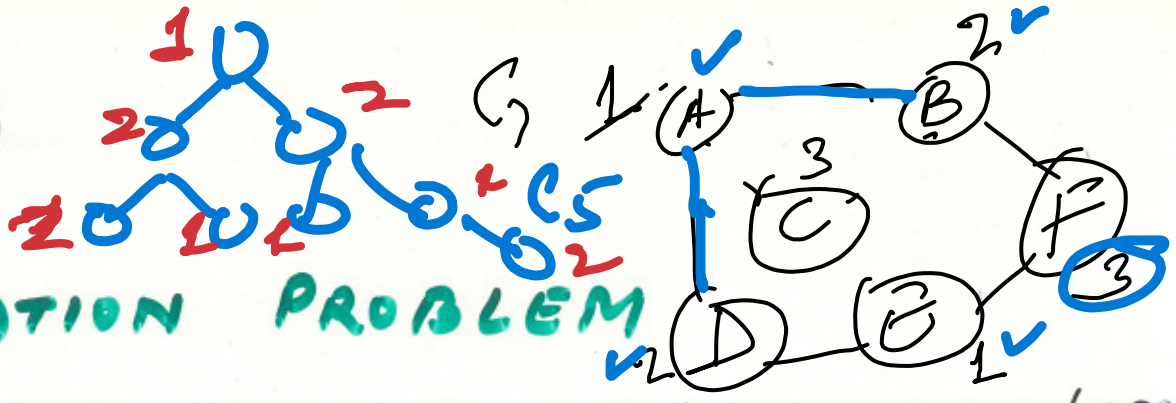
COLORING



OPTIMIZATION PROBLEM

GIVEN G, find the minimum number of colors to color G. ($\chi(G)$?)

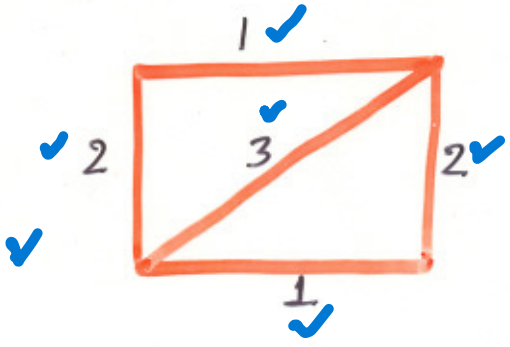
DECISION PROBLEM

GIVEN graph G and positive integer k, is $\chi(G) \leq k$?

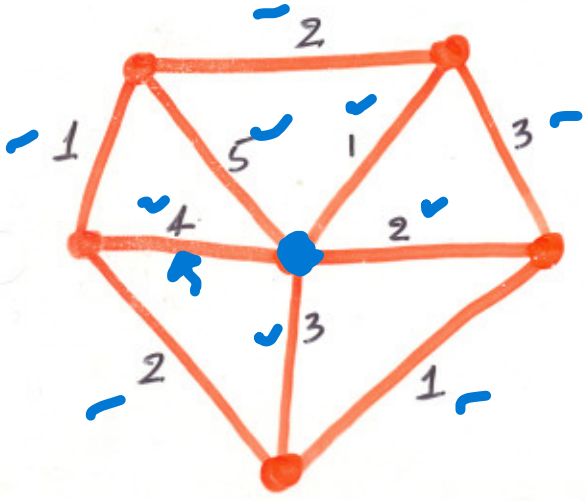
EQUIVALENCE OF OPTMIZATION AND DECISION PROBLEMS

$\chi(G)$ = CHROMATIC NUMBER OF G

EDGE COLORING



CHROMATIC INDEX = 3



Chromatic Index = 5
= highest degree $\Delta(G)$

VIZING'S THEOREM

CHROMATIC INDEX OF A GRAPH G is EITHER $\Delta(G)$ or $\Delta(G)+1$.

CLIQUE

CLIQUE$_{OPTIMIZATION}$ $\propto$ CLIQUE$_D$

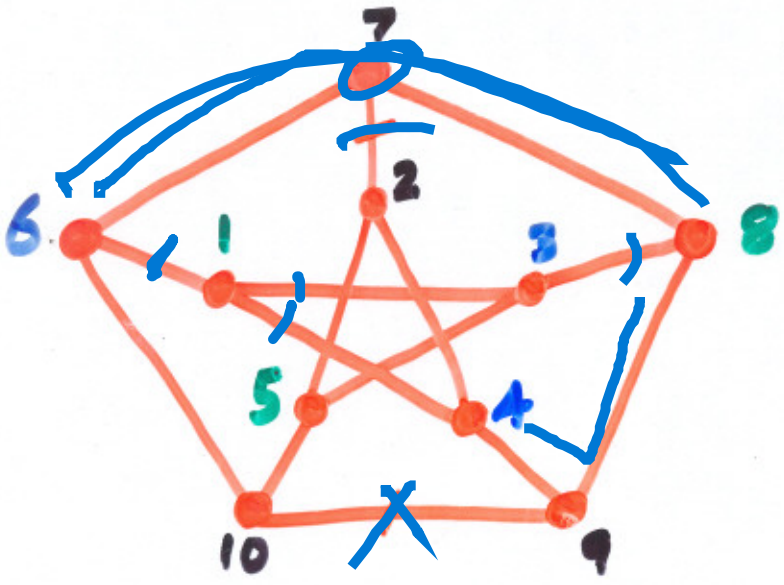
Given a graph G, find the size of the maximum clique in G.

CLIQUE$_{DECISION}$

Given G and integer k, does G have a k-clique?

CLIQUE$_{SEARCH}$ $\propto$ CLIQUE$_{OPT}$

Given G, find a maximum clique of G.



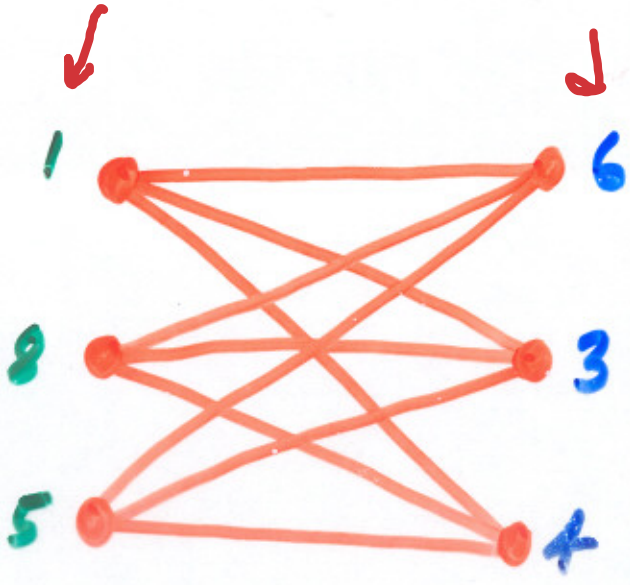
Peterson's Graph

KURATOWSKI'S THEOREM

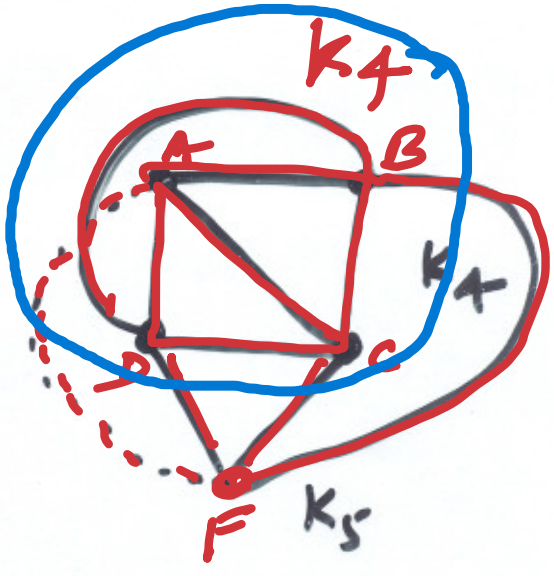
A graph is planar iff it is not homeomorphically reducible to either K_5 or $K_{3\times3}$.

✓ $G \rightarrow G - \{\{u,v\}\}$

✓ 



$K_{3,3}$



planarity testing $O(n)$

SAT (SATISFIABILITY)

Boolean Expressions

$$c_1 = x_1 + x_2 + x_3$$

$$c_2 = \bar{x}_1 + \bar{x}_2$$

$$c_3 = \bar{x}_3 + x_2 + x_4$$

$$c_4 = x_2 + \bar{x}_4$$

Satisfying Truth Assignment

$x_1 = 1$

$x_2 = 0$

$x_3 = 0$

$x_4 = 0$

$$c_1 \wedge c_2 \wedge c_3 \wedge c_4$$

$c_1 = x_1 + x_2$

$c_2 = \bar{x}_1 + \bar{x}_3$

$c_3 = x_2 + x_3$

$c_4 = \bar{x}_1 + \bar{x}_2$

$c_5 = \bar{x}_2 + \bar{x}_3$

$c_6 = x_1 + x_3$

x_1	x_2	x_3	
1	1	0	X
0	1	0	X

no satisfying truth assignment

variables: x_1, x_2, x_3

literals: $x_1, \bar{x}_1, x_2, \bar{x}_2, x_3, \bar{x}_3$

clauses: $c_1, c_2, \ldots, c_6$

SAT: Given n variables and m clauses over these variables, is there a satisfying truth assignment?

3SAT: All clauses have ~~at most~~ exactly 3 literals.

2SAT can be solved in polynomial time.

P and NP

P: Class of decision problems which have polynomially bounded algorithms

Polynomially Bounded

An algorithm whose worst-case time complexity $W(n) \leq p(n)$

n: input size

p: a polynomial such as $n^2 + 2n + 5$

Definition of P vs NP

Non Deterministic Algorithms

Graph Coloring

1. Guess a coloring $O(n)$

$$C : V \to \{1, 2, 3, \ldots, k\}$$

$C(w)$ is the color of node w

2. Verify that the guess is correct

for all $\{u, v\} \in E$ $O(m)$

$$c(u) \neq c(v).$$

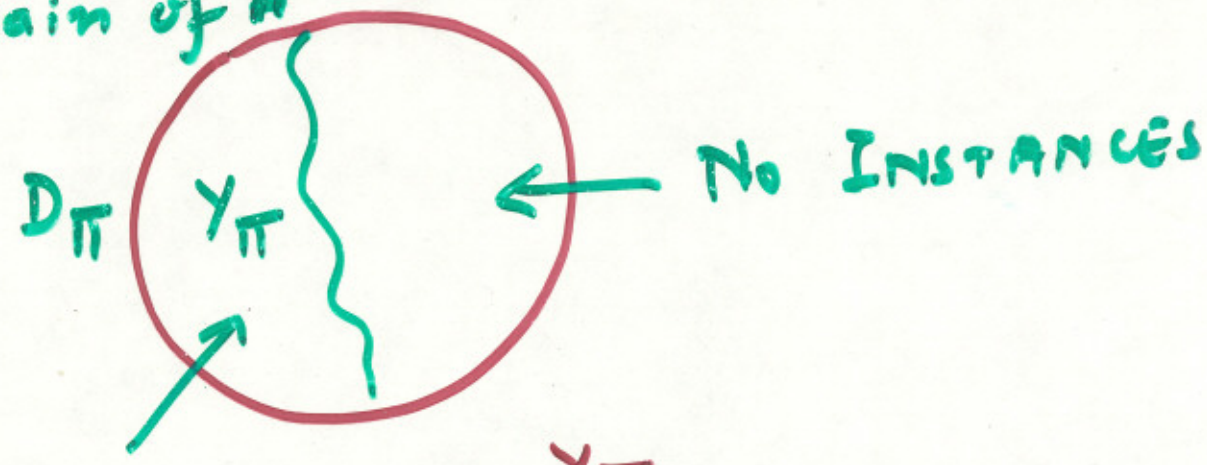
$O(n+m)$

What if one allows ability to guess correctly, if there exists a correct guess?

⇒ Non Deterministic Algo. will be

Definitions

- Problem π : Coloring
- INSTANCE I : A graph G and number of colors k.
- Domain of π



- YES INSTANCES Y_π

  $I \in Y_\pi$: if G is k-colorable
- No INSTANCES $D_\pi - Y_\pi$

  $I \notin Y_\pi$

- Polynomially Bounded Non-Deterministic algorithm is whose $w(n)$ is polynomial for all yes-instances.

NP: A class of decision problems which have polynomially bounded non-deterministic algorithms.

- NP contains those problems whose guesses are polynomial time verifiable.

- Coloring $\in$ NP
- clique $\in$ NP
- SAT $\in$ NP

guess: a truth assignment

$O(n)$ $T : \{x_1, x_2, \ldots, x_n\} \rightarrow \{0, 1\}$

check: verify that each clause is satisfied

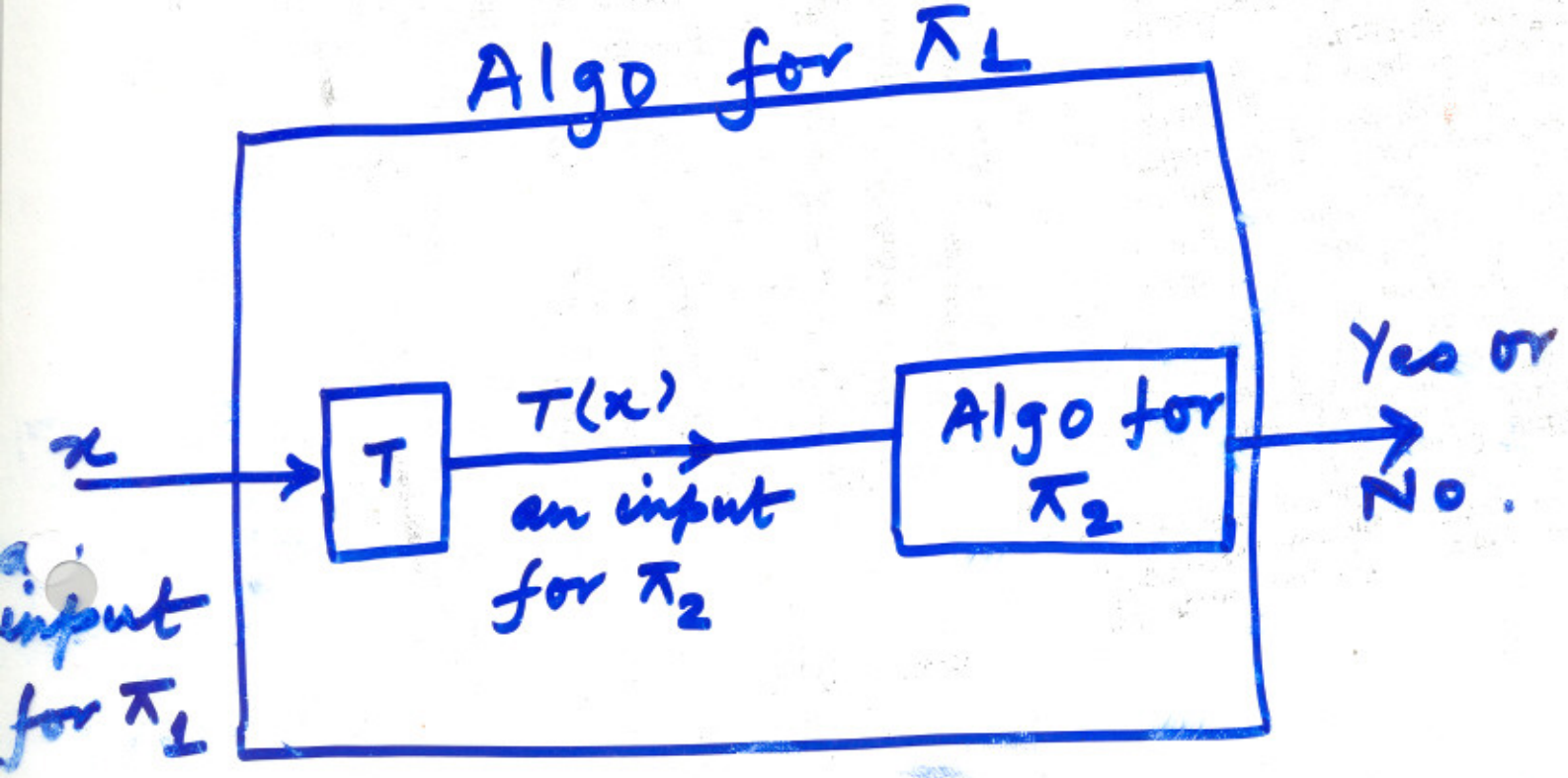
for all $c \in \{c_1, c_2, \ldots, c_m\}$

c has atleast one true literal.

$O(mn)$

$O(n+m)$

Polynomial Reduction



MST
COLOR

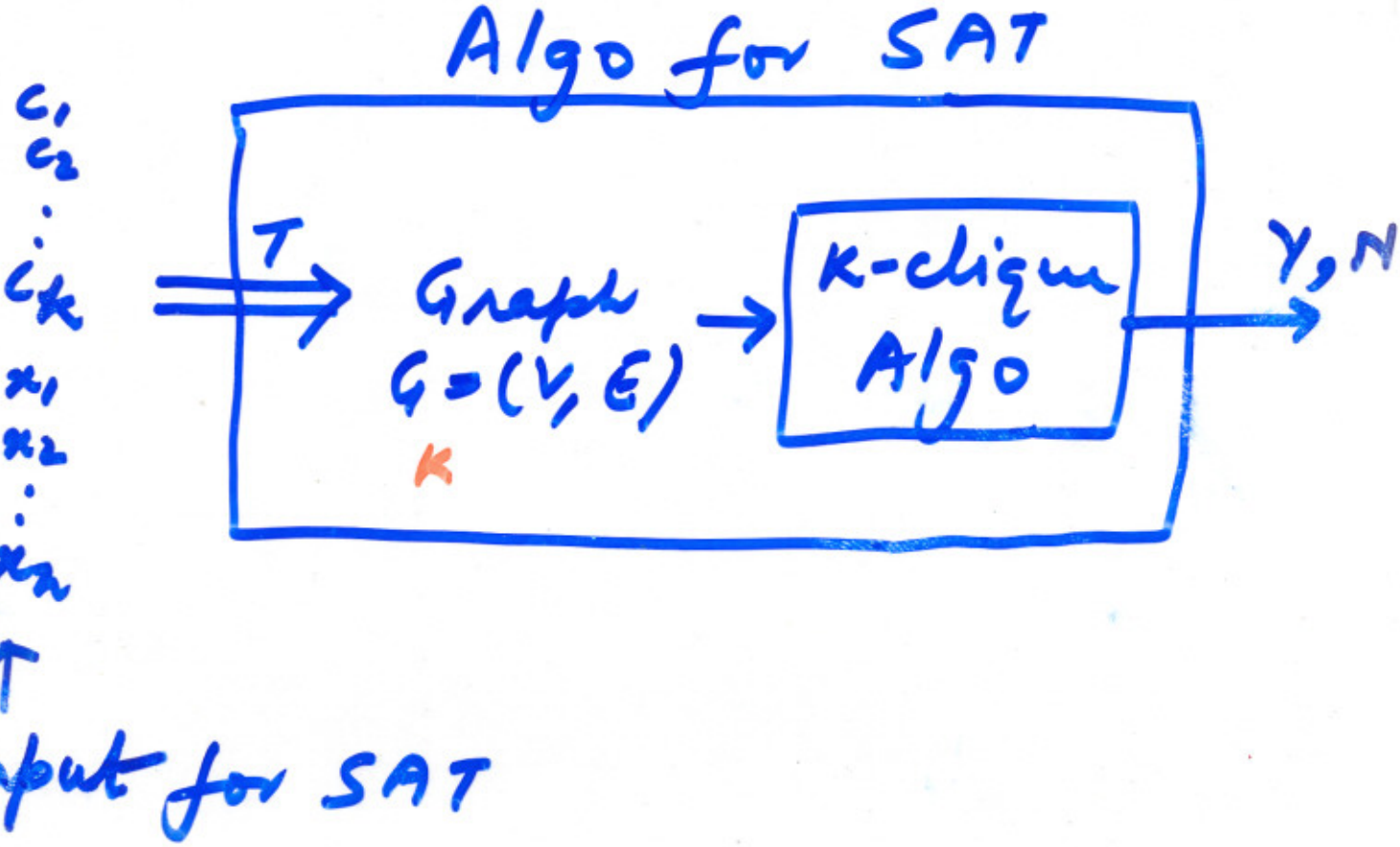
$\pi_1 \propto \pi_2$

sorting
SAT

T :1. polynomial time transformation

2. Answer Preserving

g. SAT ∝ K-Clique



↑
input for SAT

$$T:\begin{cases} V=\{\langle \sigma, i\rangle \mid \sigma \text{ is a literal in } c_i\} \\ E=\{\{\langle \sigma, i\rangle, \langle \delta, j\rangle\} \mid i\neq j \ \&\ \sigma\neq\bar{\delta}\} \end{cases}$$

clique size $K = k$, no. of clauses

T: Polyno in k & n.

$P_1 \Leftrightarrow P_3 \Rightarrow \neg P_1 \Rightarrow \neg P_2$

$C_1 = x_1 + x_2 + x_3$

$C_2 = \bar{x}_1 + \bar{x}_2$

$C_3 = \bar{x}_3 + \bar{x}_2 + x_4$

$C_4 = x_2 + \bar{x}_4$

Satisfying Truth assignment

$x_2 = 1$

$x_1 = 0$

$x_3 = \phi$

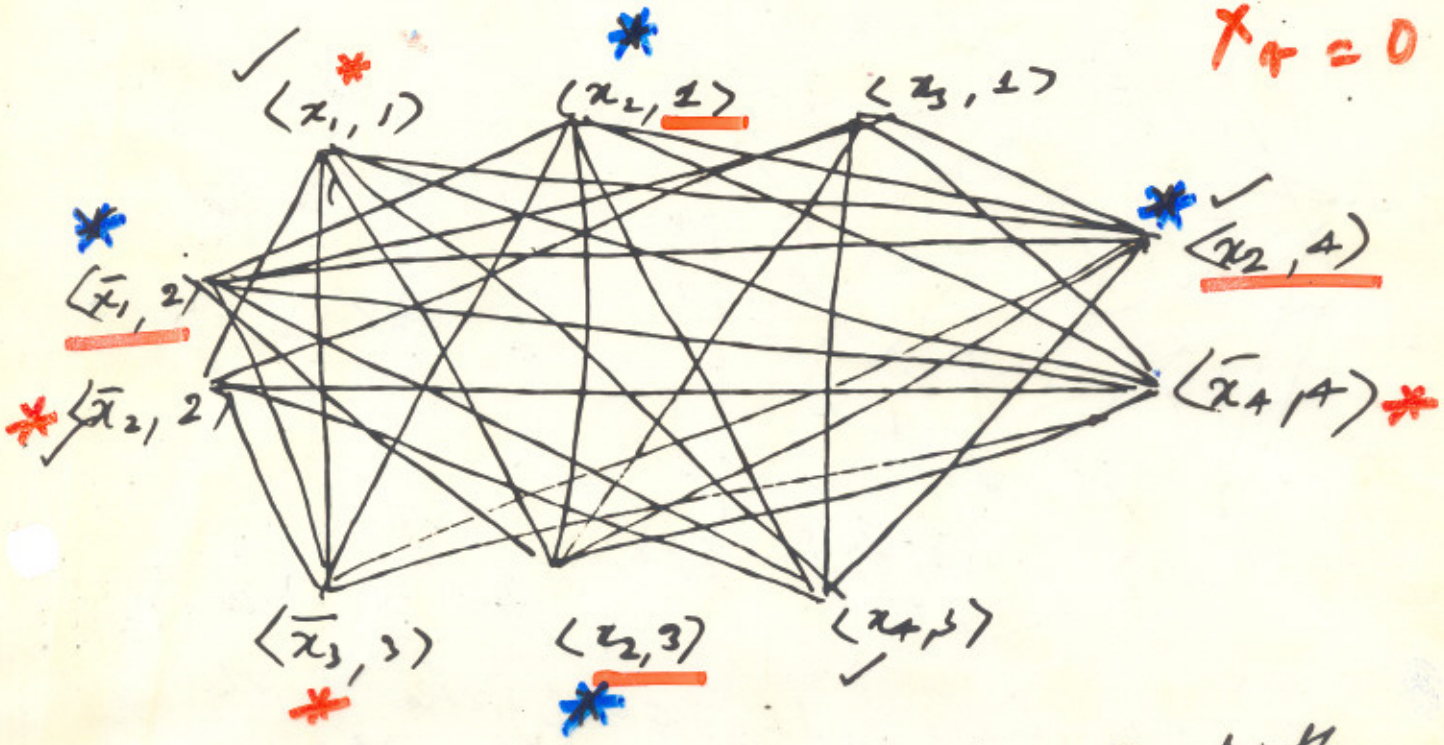
$x_4 = \phi$

$x_1 = 1$

$x_2 = 0$

$x_3 = 0$

$x_4 = 0$



4-clique? iff satisfiable truth assignment.

SAT length $O(kn \log n)$

Time for T: kn nodes

$\frac{k^2n^2}{2}$ edges

$O(k^2n^2 \log n)$

- If $\pi_1 \propto \pi_2$ and $\pi_2 \in P$ then $\pi_1 \in P$

- NP-complete (NPC)

A problem π is NP-complete

if $\pi \in NP$ ——————— (i)

and

for every other problem $\pi' \in NP$

$\pi' \propto \pi$. ——————— (ii)

- If $\pi \in NPC$, and $\pi \in P$ then $P = NP$.

Cook's theorem

$$SAT \in P \iff P = NP$$

or

If $SAT \in P \Rightarrow P = NP$

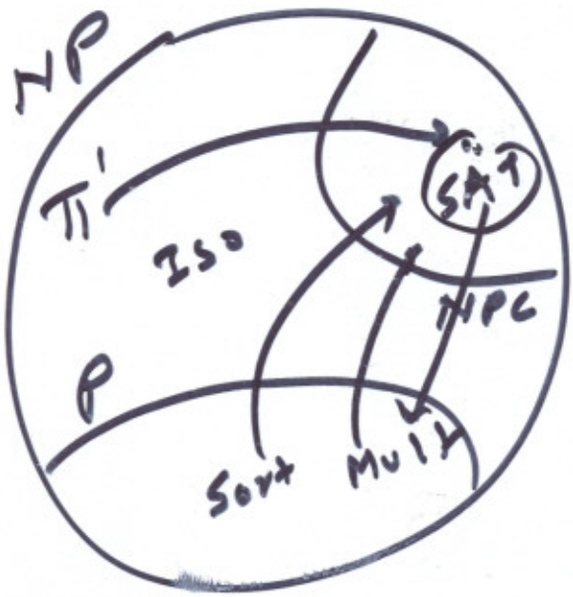
or for every problem $\pi \in NP$,
$\pi \propto SAT$.

or

$$SAT \in NPC$$

Cook's Theorem 1972

SAT ∈ NPC



$\pi' \propto$ SAT

Known Facts

1. $\pi' \in NP$.
2. π' has a poly time ND algo.
3. $W(n) \leq p(n)$

If SAT ∈ P, then P = NP

Karp
- Color — $\pi' \propto$ color; SAT $\propto$ color; color ∈ NP
- Clique
- Vertex Cover — Edge Cover
- Independent Set
- Subset Sum — $O(nC)$
- Dominating Set

Facts & Condition

To show: $\Pi' \propto SAT$

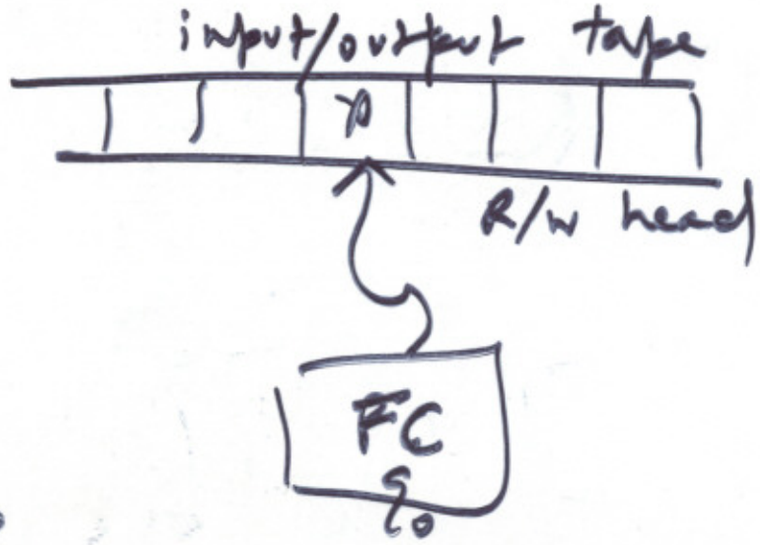
$I_{\Pi'} \xrightarrow{T} I_{SAT}$

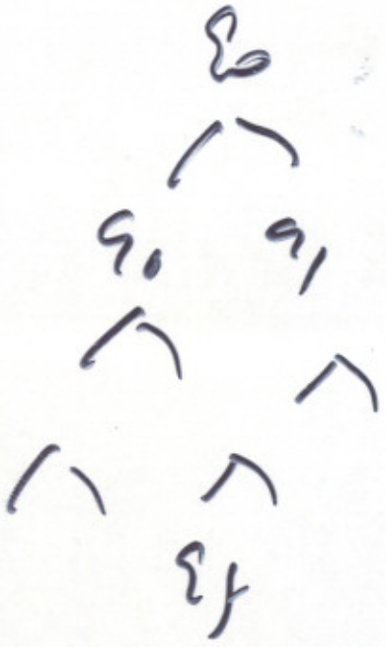
cli~~ck~~ $\Pi' \propto SAT$

1. $\Pi' \in NP$

$\Rightarrow \Pi'$ has NDTM algorithm whose time complexity $W(n) \leq p(n)$.

(5)





1. Read input char
2. Replace with another char
3. Go to another state
4. Move left/right

6. $q_0 \rightsquigarrow q_f$

Accepting Computation $\Longleftrightarrow$ Satisfying Truth Assignment

NP

SAT → COLOR

It can be shown that

(i) graph coloring $\in$ NP

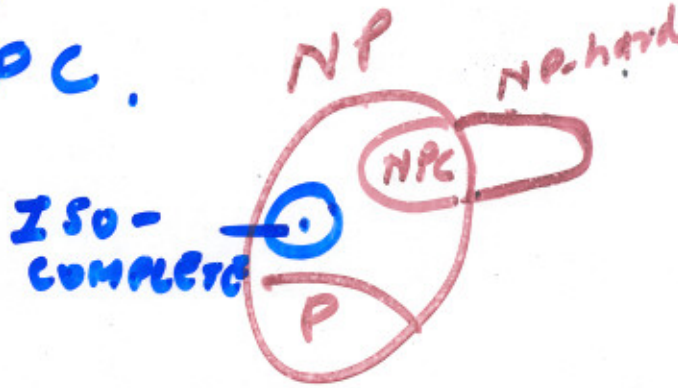
(ii) SAT $\propto$ coloring

$\Rightarrow$ graph coloring $\in$ NPC

Similarly

k-Clique, HC, TSP etc.

are also NPC.



NP-hard

π is NP-hard if for all problem $\pi' \in$ NP, $\pi' \propto \pi$.

Restriction of NPC problems

g. coloring.

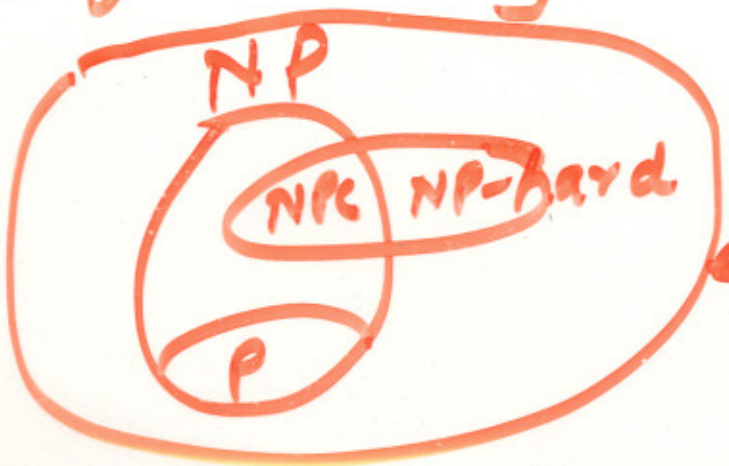
2-coloring $\in P$

3-coloring $\in$ NPC

Coloring of planar graph using 4 colors takes linear time.

3-coloring of planar graph is NPC.

Edge Coloring Δ ? $\Delta+1$



← decidable

APPROXIMATION ALGORITHMS

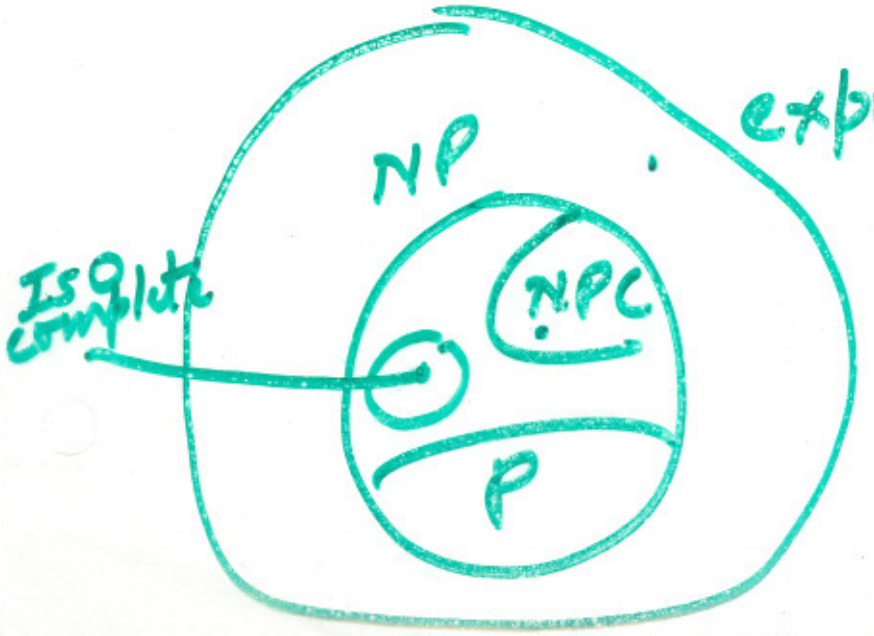
Polynomial-time algorithm for an NP-Complete (or NP-hard) problem which do not guarantee the optimal solution but would generally give one that is close to optimal.

BIN PACKING PROBLEM

Given: n objects to be placed in bins of capacity L each.

Object i requires l_i units of bin capacity.

Objective: determine the minimum number of bins needed to accomodate all n objects.



exponential

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

eg. Let $L = 10$, $l_1 = 5$, $l_2 = 6$, $l_3 = 3$, $l_4 = 7$, $l_5 = 5$, $l_6 = 4$

3 bins needed.

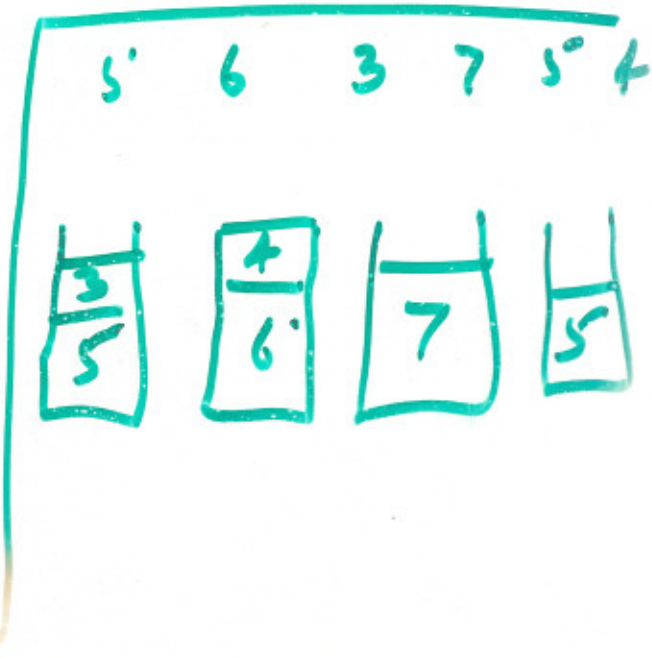
Th Bin packing problem is NP complete when formulated as a decision problem.

As an optimization problem, bin packing is NP-hard.

Approximation Algorithms for Bin Packing

1. FIRST FIT (FF)

- Label bins as 1, 2, 3, ...
- Objects are considered for packing in the order 1, 2, 3, ...
- Pack object i in bin j where j is the least index such that bin j can contain object i.



Complexity $O(n^2)$.

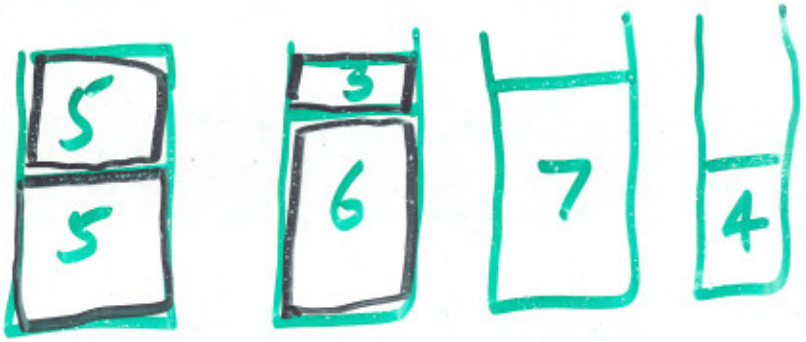
II BEST FIT (BF)

Same as FF, except that when object i is to be packed, find out that bin which after accomodating object i will have the least amount of space left.

5 6 3 7 5 4

$O(n^2)$

eg.



III First Fit Decreasing (FFD)

reorder objects so that

$l_i \geq l_{i+1}$, $1 \leq i \leq n$.

then use FF. $O(n^2)$

IV Best Fit Decreasing (BFD)

Reorder objects as above & then use BF. $O(n^2)$

Th. Packing generated by either FF or BF uses no more than $\frac{17}{10}$ OPT + 2 bins.

70% of OPT

That by either FFD or BFD uses no more than

$\frac{11}{9}$ OPT + 4 bins.

22% of OPT

$\frac{11}{9} = \frac{9}{7} + \frac{2}{7}$

1.7 OPT

Johnson 1.07 OPT

FSP $\propto$ BP

References

NP-Complete Theory, applications, examples, etc.

COMPUTER and INTRACTABILITY: A Guide to the Theory of NP-completeness

by Michael R GAREY & David S JOHNSON

Publisher: W. H. Freeman 1979

TSP MST
Triangle Inequality
Heuristic algorithm