COMPLEXITY (NP)

P = { p | p IS A DECISION PROBLEM SOLVABLE IN POLYNOMIAL TIME IN TERMS OF REPRESENTATIONS OF INSTANCES OF P USING A DET. TM }

P CONTAINS DEPTH FIRST SEARCH SOLVABLE PROBLEMS — E.G., GARBAGE COLLECTION, GRAPH CONNECTIVITY, RELATIVE PRIMALITY, MEMBERSHIP IN CFL (CKY), LINEAR PROGR. OVER REALS (NOT CONSTRAINED TO INTEGER), PRIMALITY

NP = { p | p IS A DECISION PROBLEM SOLVABLE IN POLYNOMIAL TIME ... USING A NON-DET. TM }

ALSO, NP IS

NP = { p | A SOLUTION TO AN INSTANCE OF p CAN BE VERIFIED IN DET. POLY TIME }

CAN ALSO USE NOTION OF UNBOUNDED PARALLELISM

A PROBLEM IN NP IS BOOLEAN SATISFIABILITY

CAN SOLVE WITH TRUTH TABLE BY GUESSING A SOLUTION (OR CHECK A PROPOSED ASSIGNMENT OF VALUES IN LINEAR TIME)

COMPLEXITY
OTHER CLASSES

CO-NP: COMPLEMENT OF NP

NP-HARD: SUPERSET OF NP, WHEN $P \in NP, Q \in NP\text{-HARD} \Rightarrow P \leq_p Q$

NP-COMPLETE: NP-HARD AND IN NP

QSAT IS NP-HARD BUT MAY NOT BE IN NP

PSPACE: SOLVABLE IN POLYNOMIAL SPACE

QSAT $\in$ PSPACE

EXP: PROBLEMS SOLVABLE IN EXPONENTIAL TIME

BIG QUESTION

$$P = NP?$$

IF P=NP THEN ALL NP PROBLEMS ARE IN P

IF P≠NP THEN ALL NP-C PROBLEMS AR IN EXP BUT NOT IN P

CONCRETE MODEL OF NP

CONSIDER NP DEFINED AS THE CLASS OF DECISION PROBLEMS SOLVABLE BY A NON-DETERMINISTIC TURING MACHINE IN POLYNOMIAL TIME RELATIVE TO THE LENGTH OF AN INPUT.

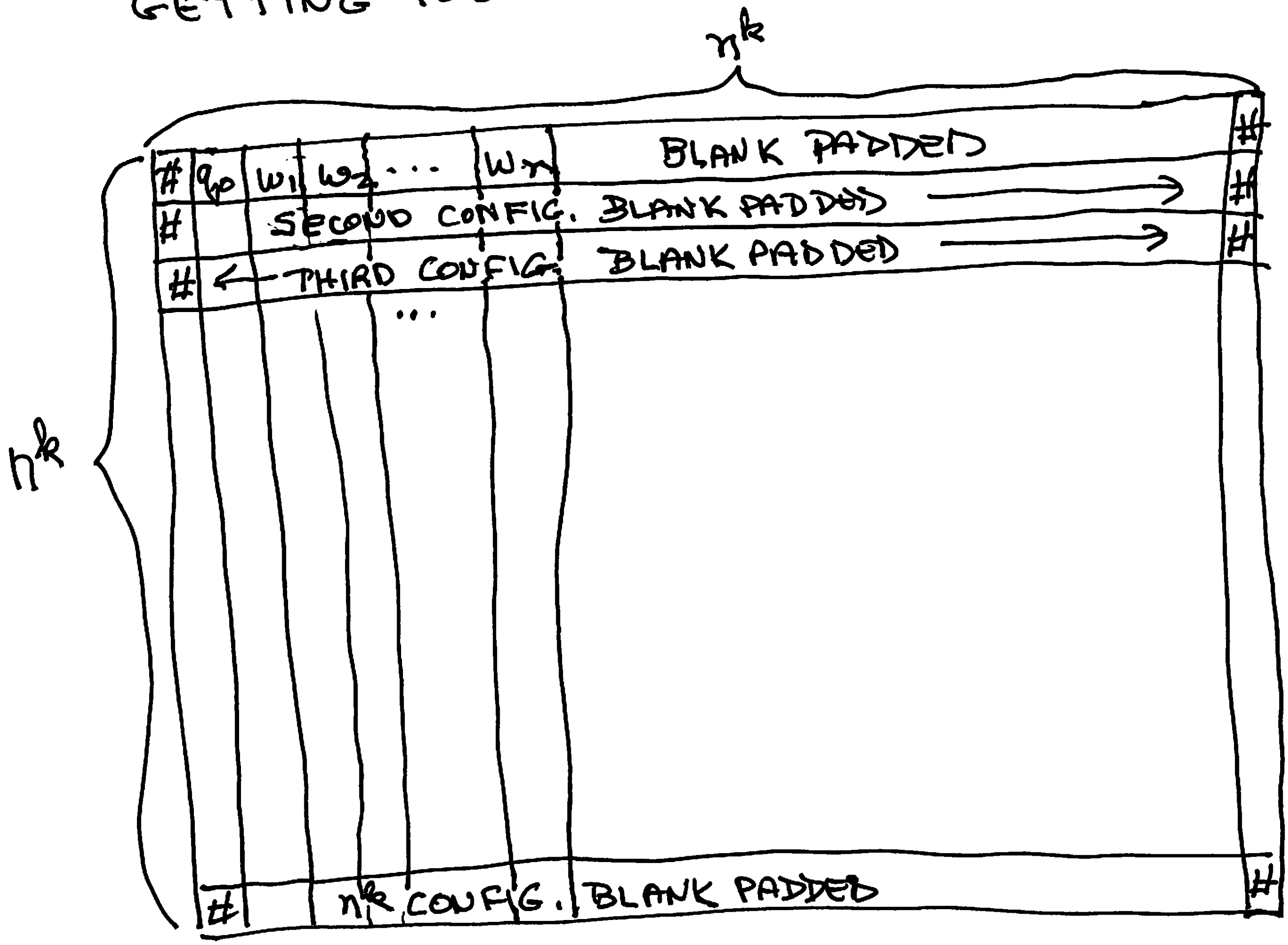
THAT IS, $A \in NP$ IFF $\exists$ NON-DET. M AND $k \geq 0$, SUCH THAT $x \in A$ IFF $x \in L(M)$ IS DECIDED IN $|x|^k$ TIME

IN OUR EARLIER WORK WE DEFINED STANDARD TURING COMPUTING AS STARTING A MACHINE TO RIGHT OF INPUT AND HAVING MACHINE COMPLETE COMPUTATION WITHOUT EVER TRAVERSING TAPE SQUARES PAST BLANK TO LEFT OF INPUT. WE COULD HAVE DESCRIBED AN EQUIVALENT MODEL THAT STARTS AT LEFT OF INPUT AND NEVER TRAVERSES LEFT OF THAT SQUARE.

ALSO, WE WROTE IDs AS MIN. LENGTH STRINGS THAT INCLUDED NO LEADING OR LAGGING BLANKS. IF WE KNOW WE NEVER TRAVERSE MORE THAN n^k CELLS, WHERE $|w|=n$, THEN WE DO NOT NEED TO EXCLUDE BLANKS LEADING OR LAGGING AND CAN JUST ALWAYS WRITE ID AS FIXED LENGTH STRING.

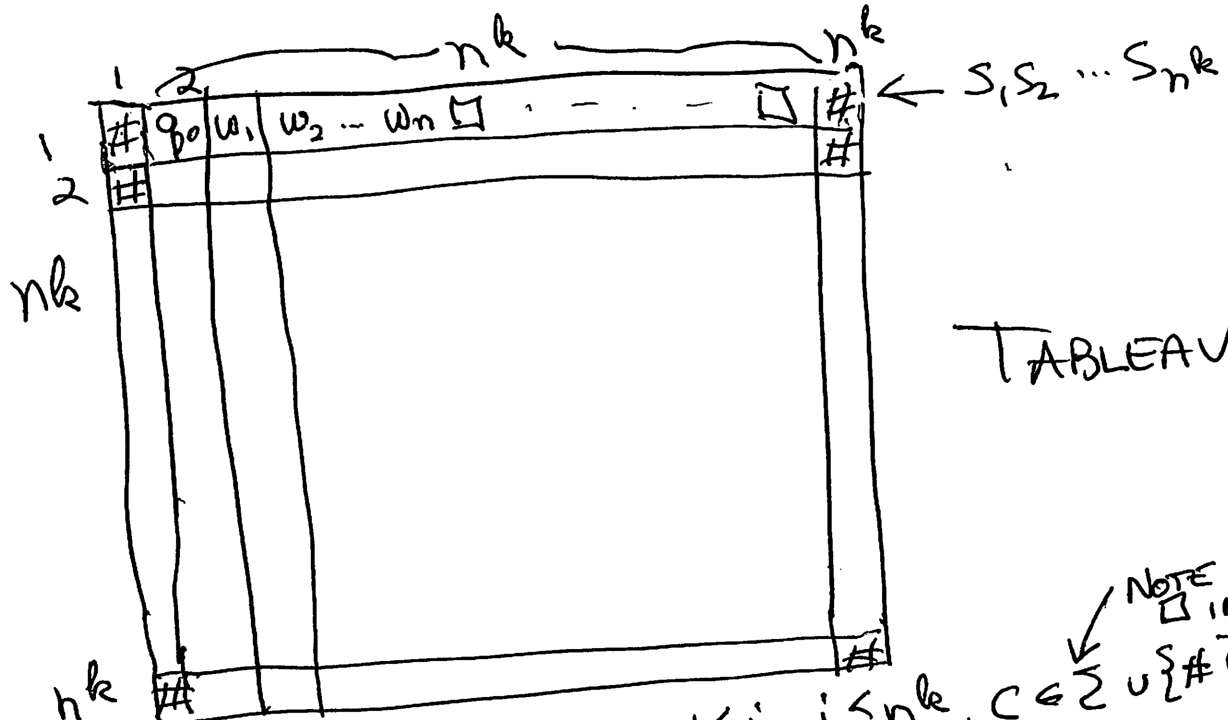
TRACES FOR AN NP PREDICATE

WE CAN MAKE TABLE (MATRIX) OF SIZE $n^k \times n^k$ TO REPRESENT TRACES OF LENGTH AT MOST n^k (WELL, WE MIGHT WANT $(n^k+3) \times n^k$ BUT THAT'S GETTING TOO FUSSY)



WANT TO SEE ACCEPTING STATE q_{acc} SHOW UP IN SOME ROW FOR SOME LEGIT TRACE

SAT AS NP-COMPLETE



NOTE CONTAINS □ IN MY WAY

BOOLEAN VARIABLES $x_{i,j,c}$ $\quad 1 \le i, j \le n^k$, $c \in \Sigma \cup \{\#\} \cup Q$

$$\phi_{\text{ENCODE}(i,j)} = \bigvee_{c \in \Sigma \cup \{\#\} \cup Q} x_{i,j,c} \wedge \bigwedge_{\substack{c,d \in \Sigma \cup \{\#\} \cup Q \\ c \neq d}} \overline{x_{i,j,c} \wedge x_{i,j,d}}$$

$$\phi_{\text{CELLS}} = \bigwedge_{1 \le i,j \le n^k} \phi_{\text{ENCODE}}(i,j)$$

ALL CELLS HAVE EXACTLY ONE SYMBOL

$$\phi_{\text{ACCEPT}} = \bigvee_{i,j} x_{i,j,\text{Acc}}$$

WHERE ACCEPT STATE IS Acc

SOME CELL INDICATES ACCEPTANCE

$$\phi_{\text{START}} = \bigwedge_{1 \le j \le n^k} x_{1,j,s_j}$$

WHERE INITIAL MARKING IS

$s_1 s_2 \cdots s_{n^k}$; NOTE $s_1 = s_{n^k} = \#$

LEGAL IS COMPLEX PART.

WE INCLUDE ALL LEGIT 2×3 WINDOWS WHICH ARE PART OF SATISFYING EXPRESSION. DEPENDS ON CONSISTENCY WITH TOP LINE AND TRANSITIONS ALLOWED IN MACHINE.

TRIVIAL EXAMPLE

q 0 R q
q 1 1 a
a 1 1 a

} ACCEPT IF A 1 (NOTE! DET.)
USE a AS ACCEPT STATE

START WITH 00100 AND SAY n^1 (DO 8 CELLS)

	1	2	3	4	5	6	7	8
1	#	q	0	0	1	0	0	#
2	#	0	q	0	1	0	0	#
3	#	0	0	q	1	0	0	#
4	#	0	0	a	1	0	0	#
5	#	0	0	a	1	0	0	#
6	#	0	0	a	1	0	0	#
7	#	0	0	a	1	0	0	#
8	#	0	0	a	1	0	0	#

#	q	0
#	0	q

$x_{1,1,\#} \wedge x_{1,2,q} \wedge x_{1,3,0} \wedge x_{2,1,\#} \wedge x_{2,2,0} \wedge x_{2,3,q}$

√

#	a	1
#	a	1

$x_{1,1,\#} \wedge x_{1,2,a} \wedge x_{1,3,1} \wedge x_{2,1,\#} \wedge x_{2,2,a} \wedge x_{2,3,1}$

#	q	1
#	a	1

$x_{1,1,\#} \wedge x_{1,2,q} \wedge x_{1,3,1} \wedge x_{2,1,\#} \wedge x_{2,2,a} \wedge x_{2,3,1}$

√

#	0	0
#	0	0

$x_{1,1,\#} \wedge x_{1,2,0} \wedge x_{1,3,0} \wedge x_{2,1,\#} \wedge x_{2,2,0} \wedge x_{2,3,0}$

√

#	0	1
#	0	1

⋮

√

#	1	1
#	1	1

√

#	1	0
#	1	0

Φ_{START} FORCES

$$\phi_{CELLS} = \bigwedge_{1 \le i,j \le n^k} \phi_{ENCODE}(i,j)$$

$$\phi_{START} = \bigwedge_{1 \le j \le n^k} X_{1,j,s_j}$$

$$\phi_{ACCEPT} = \bigvee_{1 \le i,j \le n^k} X_{i,j,q_A}$$

$$\phi_{MOVE} = \bigwedge_{\substack{1 \le i < n^k \\ 1 < j < n^k}} \phi_{LEGAL}(i,j)$$

MID POINT

$$\phi_{NDTM} = \phi_{CELLS} \wedge \phi_{START} \wedge \phi_{ACCEPT} \wedge \phi_{MOVE}$$

SATISFIABILITY (SAT)

SAT IS THE PROBLEM TO DECIDE IF AN ARBITRARY BOOLEAN EXPRESSION IS SATISFIABLE. MOST OFTEN THE BOOLEAN EXPRESSION IS IN CONJUNCTIVE NORMAL FORM (CNF)

CNF IS SPECIFIED BY A SET OF TERMS "AND"ED TOGETHER

EACH TERM IS MADE UP OF LITERALS (VARIABLES OR COMPLEMENTS OF VARIABLES) "OR"ED TOGETHER.

SAT IS IN NP AS WE CAN CHECK A SOLUTION (ASSIGNMENT OF T/F TO EACH VARIABLE) IN LINEAR TIME.

BEST KNOWN SOLUTION IS TRUTH TABLE, WHICH IS 2^N FOR N VARIABLES

SAT $\leq_p$ 3-SAT

ALL INSTANCES OF SAT CAN BE PUT IN CONJUNCTIVE NORMAL FORM (CNF)

3-SAT JUST RESTRICTS EACH DISJUNCT (CLAUSE) TO 3 LITERALS (VARIABLE OR VARIABLE COMPLEMENTED)

ASSUME WE HAVE CLAUSE

$$(t_1 + t_2 + \cdots t_n) \qquad n \geq 1$$

CASE N=1: CHANGE

(t_1) TO $(t_1 + t_1 + t_1)$

CASE N=2: CHANGE

$(t_1 + t_2)$ TO $(t_1 + t_2 + t_1)$

OR $(t_1 + t_2 + t_2)$

CASE (N=3): NO CHANGE

CASE (N>3): CHANGE

$(t_1 + t_2 + t_3 + \cdots + t_n)$

TO $\underbrace{(t_1 + t_2 + X)}_{3}\ \underbrace{(t_3 + \cdots + t_n + \overline{X})}_{N-1}$

WHERE X IS AN <u>NEW</u> VARIABLE

GROWTH IS LINEAR

SUBSET SUM

REALLY SUBMULTISET SUM

INSTANCE

$[\langle i_1, \ldots, i_n \rangle, G]$ $\quad 1 \le i_j \quad 1 \le G$

FIND SUBSET (SUBMULTISET) OF $i_1, \ldots, i_n$

THAT SUMS TO G

EXAMPLE

$[\langle 15, 17, 27, 11, 4, 12, 33, 5, 6, 21, 2 \rangle, 57]$

A SOLUTION IS 15, 17, 11, 12, 2

ANOTHER IS 17, 27, 11, 2

ETC.

BUT THERE MAY BE NO SOLUTION

BEST KNOWN TECHNIQUE IS

TRY THEM ALL, BUT THAT IS

$2^n - 1$ TRIES

THIS IS IN NP BECAUSE A PROPOSED

SOLUTION CAN BE CHECKED IN

LINEAR TIME.

3SAT $\le_p$ SUBSETSUM

DO BY EXAMPLE CONSTRUCTION

$$(a+\overline{b}+c)(\overline{a}+b+\overline{c})$$

1 PER VAR. — 1 PER CLAUSE

	a	b	c	$a+\overline{b}+c$	$\overline{a}+b+\overline{c}$
a	1	0	0	1	0
$\overline{a}$	1	0	0	0	1
b	0	1	0	0	1
$\overline{b}$	0	1	0	1	0
c	0	0	1	1	0
$\overline{c}$	0	0	1	0	1
C1	0	0	0	1	0
C1'	0	0	0	1	0
C2	0	0	0	0	1
C2'	0	0	0	0	1
G =	1	1	1	3	3

ALL BLANKS ARE 0's

PARTITION

INSTANCE

$\langle i_1, \ldots, i_n \rangle \quad i_j \geq 1$

CAN I PARTITION INTO 2 EQUAL SUBSETS THAT COVER ENTIRE SET

- IF Σi_j IS ODD, FAIL

EXAMPLE

$\langle 7, 11, 2, 5, 3, 4 \rangle$

$7 + 5 + 4 = 16$
$11 + 2 + 3 = 16$

PROBLEM IS IN NP AS CAN CHECK A SOLUTION IN LINEAR TIME

OBVIOUS SOLUTION IS ALL POSSIBLE DIVIDES

SUBSET SUM $\leq_p$ PARTITION

SUBSET SUM INSTANCE

$i_1, .., i_n, G$ LET $\Sigma = \sum_{j=1}^{n} i_j$

PARTITION INSTANCE

$i_1, .., i_n, 2\Sigma - G, \Sigma + G$

TOTAL $= \Sigma + 2\Sigma - G + \Sigma + G$
$= 4\Sigma$

FOR PARTITION CONTAINING $2\Sigma - G$ TO HAVE 2Σ

THERE MUST BE A SUBSET OF $i_1, .., i_n$ THAT EQUALS G

AND THAT LEAVES $\Sigma - G$ TO INCLUDE IN OTHER PARTITION

$2\Sigma - G + G = 2\Sigma$

$\Sigma + G + \Sigma - G = 2\Sigma$

NOTE! $2\Sigma - G$ & $\Sigma + G$ CANNOT BE IN SAME PARTITION AS THAT WOULD HAVE AT LEAST 3Σ

$2\Sigma - G + \Sigma + G = 3\Sigma$

OPTIMIZATION OF SUBSET SUM PARTITION

SUBSET SUM IS TO GET AS CLOSE TO G AS POSSIBLE WITHOUT EXCEEDING

PARTITION IS TO GET AS CLOSE AS POSSIBLE TO EQUAL

GRAPH THEORY PROBLEMS

UNDIRECTED ACYCLIC GRAPHS

1. COLORING PROBLEM

2. INDEPENDENT SET PROBLEM

3. VERTEX COVERING PROBLEM
   (REALLY ABOUT EDGES)

VERTEX COVERING

CHOOSE SET OF VERTICES SUCH THAT EVERY EDGE HAS AN ASSOCIATED SELECTED VERTEX

OPTIMIZATION VERSION

WHAT IS MINIMUM SIZE SET?

DECISION PROBLEM

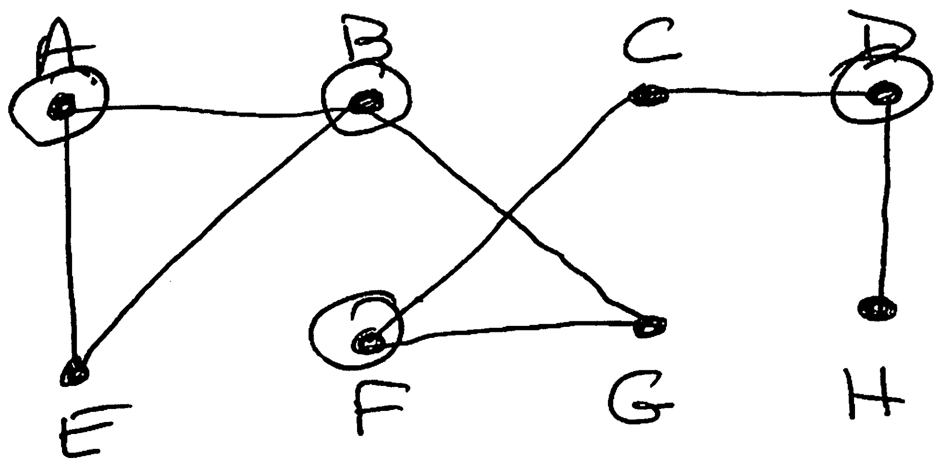
GIVEN $k > 0$, CAN THE GRAPH BE COVERED BY K VERTICES

CAN SOLVE OPTIMIZATION IN POLY TIME IF CAN SOLVE DECISION IN POLY

TRY $1, 2, \ldots, n$ = # VERTICES

NOTE: n ALWAYS WORKS

VC K=4



A, B, F, D COVERS

OR

E, B, F, D COVERS

INDEPENDENT SET

CHOOSE SET OF VERTICES
SUCH THAT NO TWO VERTICES
HAVE COMMON EDGE (ARE CONNECTED)

OPTIMIZATION VERSION

WHAT IS MAXIMUM SIZE SET?

DECISION PROBLEM

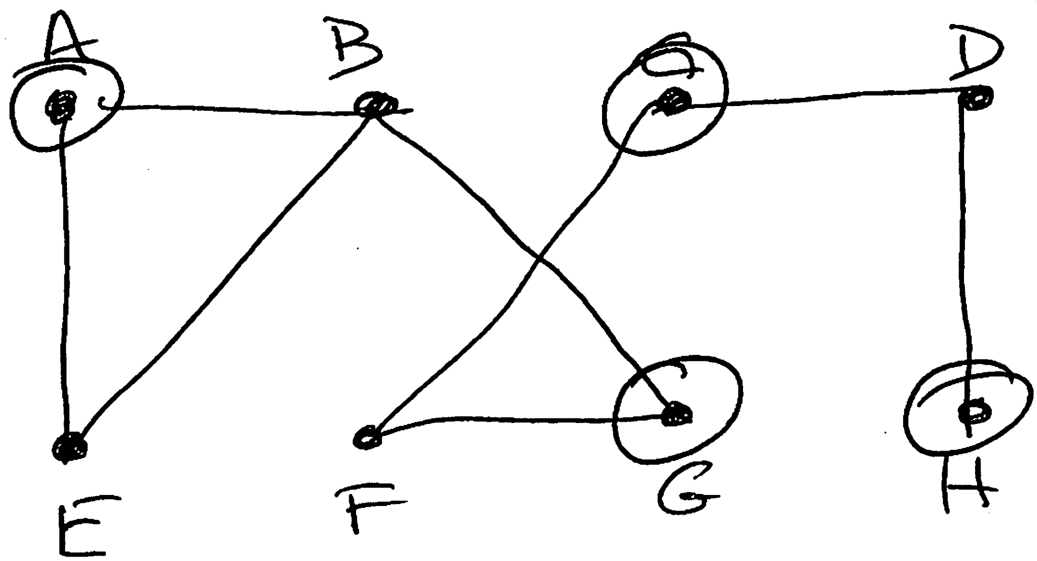
GIVEN $k > 0$, CAN THE GRAPH ~~BE~~
HAVE AN INDEPENDENT SET OF
SIZE k

CAN SOLVE OPTIMIZATION IN POLY TIME
IF CAN SOLVE DECISION IN POLY

TRY $n, n-1, \ldots, 1$ $n = \#$ VERTICES

NOTE! 1 ALWAYS WORKS

IS $k=4$



A G C H is IS

OR E G C H is IS

GRAPH COLORING

COLOR EACH VERTEX SO NO NEIGHBOR VERTEX HAS SAME COLOR

OPTIMIZATION

WHAT IS MINIMUM # OF COLORS

DECISION

GIVEN $k>0$, CAN WE COLOR WITH k COLORS

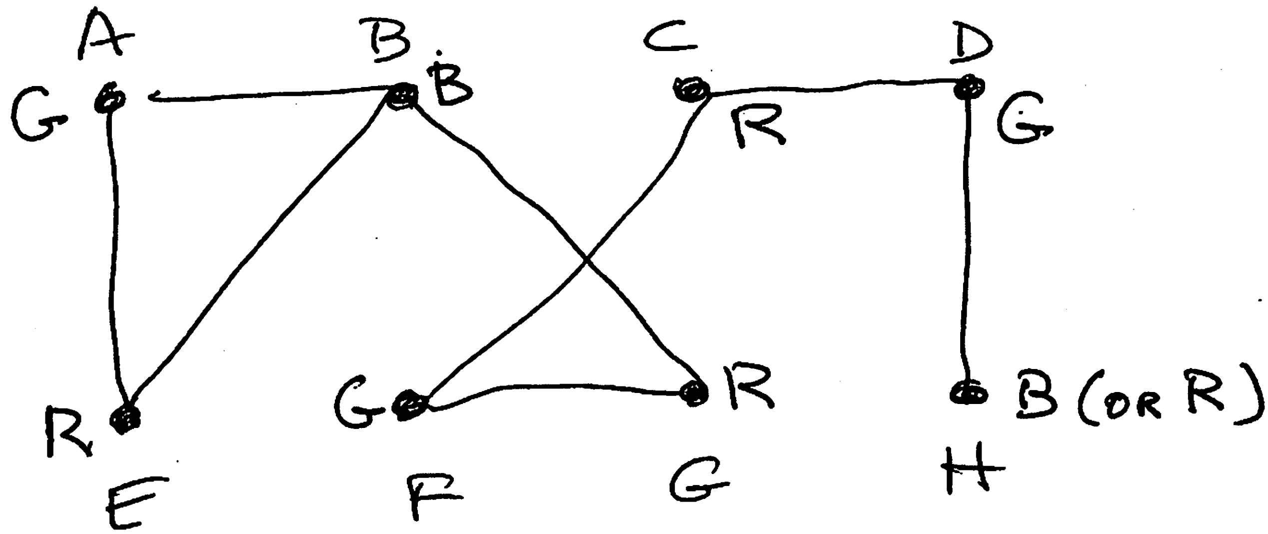
CAN SOLVE OPTIMIZATION IN POLY IF DECISION SOLVABLE IN POLY

TRY $1, 2, \ldots, n = \#$ VERTICES

n ALWAYS WORKS

4 ALWAYS WORKS IF PLANAR (NO CROSSING EDGES)

GC k=3



ABOVE IS ONE OF MANY 3-COLOR SOLUTIONS