· · · · · · · ·	March 2025	Boolean	Satisficbility	P, and NP
n n n N Pl	CM · · · · · · · · · · · · · · ·			· · · · · · · · · ·
· · · · · ·	Boolean Satisfiability	Problem	· · · · · · · · · · ·	· · · · · · · · · · ·
 <b>X</b>	Annonneements			· · · · · · · · · ·
× ×	Defining P vs. NP	· · · · · · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · · · ·
				· · · · · · · · · ·
· · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · · · · ·
· · · · ·	· · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · ·	· · · · · · · · · ·	· · · · · · · · · ·
· · · · ·	· · · · · · · · · · · · · · · · · ·		· · · · · · · · · ·	· · · · · · · · · ·
· · · · ·	· · · · · · · · · · · · · · · · · ·		· · · · · · · · · ·	· · · · · · · · · · ·
		· · · · · · · · · · · ·		· · · · · · · · · ·

Systems of Boolean Equations
$\begin{array}{llllllllllllllllllllllllllllllllllll$
<u>CNF-SAT</u> (i.e. AND-of-OR SAT)
$ \begin{array}{c} x_1 & \sqrt{7} & x_2 \\ \hline \\ x_5 & \sqrt{x_1} & \sqrt{x_6} \\ \hline \\ x_3 & \sqrt{7} & x_6 \\ \hline \\ x_3 & \sqrt{7} & x_6 \\ \hline \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_5 \\ \hline \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_6 \\ \hline \\ \end{array} \begin{array}{c} x_6 & \sqrt{7} & x_6 \\ \hline \\ \end{array} \end{array}$

Systems of Boolean Equations
$\begin{array}{rcl} \underline{XOR} - \underline{SAT} & \underbrace{\mbox{Polynomial}} & -\underline{time} & \underline{Solvev} & (\underline{via} & \underline{Gaussian} \\ & \underline{Etruination} & \\ & \underline{X_1} & \underline{\oplus} & \underline{X_2} & \underline{\oplus} & \underline{X_3} \\ & \underline{X_5} & \underline{\oplus} & \underline{X_1} & \underline{\oplus} & \underline{X_6} \\ & \underline{X_3} & \underline{\oplus} & \underline{\neg X_6} & \underline{\oplus} & \underline{\neg X_5} \end{array} \end{array} $
CNF-SAT (i.e. AND-of-OR SAT)
$ \begin{array}{c} x_1 & \sqrt{7}x_2 & \sqrt{x_3} \\ \hline \\ x_5 & \sqrt{x_1} & \sqrt{x_6} \\ \hline \\ x_3 & \sqrt{7}x_6 & \sqrt{7}x_5 \end{array} \end{array} \begin{array}{c} \text{Does there exist an assignment} \\ \hline \\ \hline \\ \hline \\ x_7 & = \overline{\alpha} \in [20,1]]^n \\ \hline \\ \text{s.t. every clause is satisfied ?} \end{array} $

Systems of Boolean Equations
$\begin{array}{rcl} \underline{XOR-SAT} & \underset{X_{1}}{\overset{(Via)}{\longrightarrow}} & \underset{Z_{2}}{\overset{(Via)}{\longrightarrow}} & \underset{Z_{3}}{\overset{(Via)}{\longrightarrow}} & \underset{X_{5}}{\overset{(Via)}{\longrightarrow}} & \underset{Z_{3}}{\overset{(Via)}{\longrightarrow}} & \underset{Z_{3}}{\overset{(Via)}{\overset{(Via)}{\longrightarrow}} & \underset{Z_{3}}{\overset{(Via)}{\longrightarrow}} & \underset{Z_{3}}{\overset{(Via)}{\longrightarrow}} & \underset{Z_{3}}{\overset{(Via)}{\longrightarrow}} &$
CNF-SAT (i.e. AND-of-OR SAT)
$ \begin{array}{c} x_1 & v & \tau x_2 & v & x_3 \\ \hline \\ x_5 & v & x_1 & v & x_6 \\ \hline \\ x_3 & v & \tau x_6 & v & \tau x_5 \end{array} \end{array} \begin{array}{c} T \\ T $
No known efficient algorithm!

· ·		) ) 	, \ a	201	· · ·		( <u>C</u>	. N	F	)	•	Fe		<u>m</u>	<u>u</u>	<u>a</u>			•	•		•	•		•	• •	•			· ·	•	•	•	•		•	•	•	•	· ·
	•	· · · · · ·	•		×,	N N		י ר א ר		. \		. >	3′	, , , , ,	· · · ·	· · · · · · · · · · · · · · · · · · ·		X	۲V	n V		, ×	•		•		, ) , )				×3		J	ر <i>ا</i>	×د			<u>ر</u> ک	< 5 /	
· ·	•	•	•	• •	• •	•	•	•	•	•	•	•	•	•	••••			•	•	•	•	•	•	•	•	• •	•	•	•	· ·	•	•	•	• •	• •	•	•	•	•	• •
• •	•	•	•	• •		•	•	•	•	•	•	•	•	•	• •			•	•	•	•	•	•	•	•	• •	•	•	•		•	•	•	•		•	•	•	•	• •
· ·	•	•	•	• •	• •	•	•	•	•	•	•	•	•	•	• •		•	•	•	•	•	•	•	•	•	••••	•	•	•	••••	•	•	•	• •	• •	•	•	•	•	
	•	•	•	• •	• •	•	•	•	•	•	•	•	•	•	• •		•	•	•	•	•	•	•	•	•	• •	•	0 0	•	• •	•	•	•	• •	• •	•	•	•	•	
•••	•	•	•	• •	· ·	•	•	•	•	•	•	•	•	•	· ·		•	•	•	•	•	0	•	•	•	• •	•	•	•	· ·	•	•	•	• •	· ·	•	•	•	•	• •
	•	•	•	o -	• •	•	•	•	•	•	•	•	•	•	• •				•	•	•	•	•	•	•	• •	•	0	•	• •	•	•	•	• •	• •	•	•	•	•	
••••	•	•	0 0	• •	• •	•	•	•	0	•	•	•	•	•	• •		•	•	•	•	0	0 0	•	•	•	• •	•	•	•	• •	•	•	0	• •	• •	•	•	•	•	• •
	0	•	•	•	• •	•	•	•	•	•	•	•	•	•	• •		•		•	•	•	•	•	•		• •	•	0	•	• •	•	0	•	• •		•	•		•	• •
••••	0	•	•	•	• •	•	•	•	0	•	•	•	e 0	•	• •		•	•	•	•	0	÷	•	•	•	• •	•	0	•	• •	•	•	0	• •	• •	•	•	•	•	• •
• •	•	•	•	• •			•	•	•	•	•	•	•	•	• •			•	•	•	•	•	•	•	•	• •	•	•	•	• •	•	•	•	• •		•	•	•	•	• •
· ·	•	•	•	•	• •	•	•	•		•	•	•		•	· ·		•	•	•	•	•	•	•	•	•	· ·	•	•	•	· ·	•	•	•	• •	· ·	•	•		•	· ·

· · ·		)00	10		~ . ~ .	. (	<u>(</u> <u>C</u>	21	F.	)	•	Fe	5r	<u>m</u>	<u>ر (،</u>	~ ~	•	•	•	•	•	•	• •	•	•	•	•	· · ·	•	•	•	• • • •	•	•	•	· · ·	•	•	· · ·
	• • •	· · /.	•			н н М		י ז ג ז	2	. \		. ×		 		•		XS		7	· · >	K,	  	, √ 2 ,	.>	ہ ر می ک	) )	, , , , , , , , , , , ,		×	3			×ږ	• • •			×s	n n h h h
• •	•	•	•	• •	•	•	•	•	•	•	•	•	•	• •	•	•	•	•	•	•	•	•	• •	•	•	•	•	•••	•	( N				c~	ن		S	•	• •
· ·	•	•	•	· · ·	•	•	•	•	•	•	•	•	•	• • • •	•	•	•	•	•	•	•	•	• • • •	•	•	•	•	· · ·	•	· · · ·	Κ,	., X ., X	.2.			· . ·	/ ·	X	· · ·
· ·	•	•	•	••••	•	•	•	•	•	•	•	•	•	• •	•	•	•	•	•	•	•	•	• •	•	•	•	•	• •	•	•	•	• •	•	•	•	••••	•	•	· ·
· ·	•	•	•	· ·	•	•	•	•	•	•	•	•	•	• •	•	•	•	•	•	•	•	•	• •	•	•	•	•	· · ·	•	•	•	• •	•	•	•	· · ·	•	•	· · ·
· ·	•	•	•	••••	•	•	•	•	•	•	•	•	•	• •	•	•	•	•	•	•	•	•	• •	•	•	•	•	••••	•	•	•	• •	•	•	•	· ·	•	•	· ·
• •	•	•	•	• •	•		•	•	•	•	•	•	•	• •	•		•	•	•	•	•	•	• •	•	•		•	• •	•	•	•	• •	•	•	•	• •	•	•	• •
· ·	•	•	•	• •	•	•	•	•	•	•	•	•	•	• •	•	•	•	•	•	•	•	•	• •	•	•	•	•	• •	•	•	•	• •	•	•	•	· ·	•	•	· ·
• •	•	•	•	• •	•	•	•	•	•	•	•	•	•	• •	•	•	•	•	•	•	•	•	• •	•		•	•	• •	•	•	•	• •	•	•	•	••••		•	• •

· ·	<u>B</u>	<u> </u>	e cu	· · · · · · · · · · · · · · · · · · ·		<u> </u>	NF	- )	•	Fe		<u>n</u> c	r (0		•	· · ·	•	•	•	· · ·	•	•	••••	•	•	•	• •	•	•	• •	•	•	· ·	•	· · ·
	)	· · ·		×,,	V	. –	۲ <u>۲</u>	-	V.	. >	· · ·	) )							, ×	· · ·	↓ . ↓	, , X	ہ مر م	./			×z		J ·	י ק ר י	ر م			×s	
· · ·		· ·	•	· ·	•				7.0		  	•		•	•	· ·	•	•		· ·	•	•	· · ·		•		۲ ۱		av	íc		لع	S	•	· ·
• •	•	• •	•	• •	•	•	. V . V	an ts		nec	2	e tu	er.	•	•	••••	•	•		• •		•	• •	•	•	•	· X · ·	( )	X2			- · _ ·	· / ·	× .	· ·
· · ·	• •	· ·	•	· · ·			· · ·	•		•	J .  	•		•	•	· ·	•	•		· ·		•	· ·		•		· ·	•	•	· ·		•	· ·	•	· · ·
· ·		· ·	•	· ·	•		· · ·	•	•	•	· ·		•	•	•	· ·	•	•	•	· ·		•	••••	•	•	•	• •	•	•	 	•	•	· ·	•	· ·
· ·	• •	• •	•	· ·	•	•	• •		•	•	•••	•	•	•	•	• •	•	•	•	••••	•	•	••••	•	•	•	• •	•	0 0	• •	•	•	• •	•	• •
· ·	• •	· ·	•	· ·	•		• •	•	•	•	· ·		•	•	•	· ·	•	•	•	· ·		•	· · ·	•	•	•	• •	•	•	· ·	•	•	· ·	•	· ·
· ·	• •	· ·	•	· ·	•		· ·	•	•	•	· ·		•	•	•	· ·	•	•	•	· ·		•	· ·	•	•	•	• • • •	•	•	• • • •	•	•	· ·	•	· ·
• •	• •	• •	•	• •	•	•	• •	•	•	•	• •	•	•	•	•	• •	•	•	•	• •	•	•	• •	•	•	•	• •	•	•	• •	•	•	• •	•	• •

· ·	B	001	em	· · ·	, , ,	CN	IF)	) . ) .	Fe	or y	<u>nu</u>	<u>la</u>	•	· ·	•	· ·	• •	· ·	· ·	· ·	•	· •	· ·	· ·	· · ·	•	· ·	· ·	•
	· · · ·	· · ·		×		· · · · · · · · · · · · · · · · · · ·	۲ <u>ې</u>	  	· · · ·		) } }	· · · · · · · · ·		XS			· · ·		م م X				δ δ δ δ δ δ δ δ δ δ δ δ δ δ	, , , , , , , ,	1 X 6 1		· · ·	×s	) .
· · · · · · · · · · · · · · · · · · ·	•	· · · · · · · · ·	· · ·	· · · · · · · · · · · · · · · · · · ·	•		vov its		ura - 61 Nec		e e e	· · ·		· · · · · · · · · · · · · · · · · · ·				L L L L L L L L L L L L L L L L L L L		· · ·	•	η η η η η η η		arv ( X2.	در اخ ،	الع 	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	
· · ·		· · ·	· · ·	· · ·		· · · · · · · · · · · · · · · · · · ·	• • • •	· · ·		J .   		· · ·		(ì	a. .e. Col	lec	S)R Fr		120 06		a 1:fe	vaQ	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · ·		· · ·	· · ·	· · ·
· · ·	•	· · ·	· ·	· · ·	•	· · ·	•	· · ·	•	· · ·	•	· · ·	•	· · ·		· · ·	· ·	· · ·	· · ·	· · ·	•	· · ·	· · ·	· · ·	· · ·	• • • •	· · ·	· · ·	•
· · ·	•	· · ·	· · ·	· · ·	•	· · · · · · · · · · · · · · · · · · ·	•	· · ·	•	· · ·	• • • •	· · ·	•	· · ·		· · ·	· · ·	· · ·	· · ·	· · ·	•	· · ·	· · ·	· · ·	· · ·	• • • •	· · ·	· · ·	•
• •	•	· ·	• •	· ·	•	· ·	•	• •	•	• •	•	••••	•	· ·	•	• •	• •	• •	• •	· ·	•	· •	· ·	• •	· ·	•	• •	•••	•

Boolean (C	NF) Formula		
$ = \left( \times_{1} \vee \cdots \right)^{n} $	$T \times_2 \times X \times_3 ) \land ($	$(X_{5} \times X \times X_{1} \times X_{2})$	$ \bigwedge \left( \begin{array}{c} X_3 \\ X_3 \\ \end{array} \right) \neg X_6 \\ \swarrow \\ \neg X_5 \\ \end{matrix} \right) $ $ \left( \begin{array}{c} X_3 \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\$
Conjunction (i.e. AND) of Clauses	its negation	m clauses: a disjunction (i.e. OR) of collection of	a I:tevals
		.       .	.       .
		.	.       .

Boolean (CNF) Formula	•
SAT. Given a boolean formula $\phi$ , does there exist a satisfying assignment?	· · · · · · · · · · · · · · · · · · ·

Boolean (CNF) Formula
$\oint = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_5 \vee x_1 \vee x_6) \wedge (x_3 \vee \neg x_6 \vee \neg x_5)$
The formula: Formula: Tormula: Conjunction (i.e. AND) of Clauses Collection of literals
SAT. Given a boolean formula \$, does there exist a satisfying assignment?
That is, $\bar{\chi} = \bar{\alpha} \in \mathcal{E}[1]$ set $\Phi(\bar{\alpha}) = 1$

•	· · ·	Þ	, , , ,			X				, , ,	< <u>2</u>	-	$\checkmark$	•		Χ.	3	)	•				,,	ר )	X 2	· · · ·		· · ·		×,		• • •				X X	, `3		· · · ·	Х,		). ).	•	•	•	•	•	•
•		1	<u></u> S	•	. (	þ þ	•		50	×+	-ر <u>د</u>	\$	-(0	- (		Q		7 ,	•	•	•	•	•	· · ·	•	•	•	•	•	•	•	•	•	•	• •			•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	· · ·	•	•	•	•	•	•	•	•	•	•	• •			•	•	•	•	•	•	•	•	•	•	•
•	•	•	•					•	•	•	•			•	•	•	•	•		•		•		· · ·		•			•	•	•	•		•	· ·	•		•		•		•			•		•	•
•	•		•	•	•	•	•	•	•	•	•			•	•	•	•		•	•	•	•		· · ·	•	•	•	•	•	•	•	•	•	•	• •	•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	, ,	•	•	•	•	•	•	•	•	•			•	•	•	•	•	•	•	•	•	•		•	•	•		•	•	•	•	•	•	· ·			•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	· · ·	•	•	•	•	•	•	•	•	•	•	• •		•	•	•	÷	0	•	•	•	0	•	•	•
•	•	• •	•	•	•	•	•	•	•	•	•	•		•	•	•	•	•		•	•	•	•	· · ·	•	•	•	•	•	•	•	•	•	•	· ·			•		•	•	•		•	•	•	•	•
•	•	2	•			•	•	•	•	•	•			•	•	•	•	•		•	•	•		· · ·	•	•			•	•	•	•	•	•	• •			•		•		•		•	•		•	•
•	•		•	•	•	•	•	•	•	•	•			•	•	•	•	•	•	•	•	•	•	· · ·	•	•	•	•	•	•	•	•	•	•	• •			•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•			•	•	•	•				•	•		· · ·	•	•		•	•	•	•	•	•	•	• •	•	•	•	•	•	•	•	•	•	•	•	•	

$ \varphi = (X_1 \vee X_2 \vee \neg X_3) \wedge (\neg X_2 \vee \neg X_1) \wedge (X_3 \vee X_1) $
Is & satisficble?
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$

$ \varphi = (X_1 \vee X_2 \vee \neg X_3) \wedge (\neg X_2 \vee \neg X_1) \wedge (X_3 \vee X_1) $
Is & satisficble?
$X_1 X_2 X_3 \qquad C_1 C_2 C_3 \qquad \qquad$
• • • • • • • • • • • • • • • • • • •

$ \Phi = \left( X_{1}^{*}, V_{2}^{*}, X_{2}^{*}, V_{3}^{*}, X_{3}^{*} \right) \wedge \left( X_{1}^{*}, V_{2}^{*}, X_{2}^{*}, V_{3}^{*}, X_{3}^{*} \right) \wedge \left( X_{1}^{*}, V_{2}^{*}, X_{3}^{*}, V_{3}^{*}, X_{3}^{*} \right) \wedge \left( X_{1}^{*}, X_{2}^{*}, V_{3}^{*}, X_{3}^{*} \right) \wedge \left( X_{1}^{*}, X_{3}^{*}, X_{3}^{*} \right) \wedge \left( X_{1}^{*}, X_{3}^{*} \right) \wedge \left( X_{1}^{*} \right) \wedge \left( X_{1}^{*} \right) \wedge$	- Τ Χ <sub>2</sub>	<pre></pre>			× 3	, , , , , , , , , , , , , , , , , , ,		· ·	· ·	· ·
Is & satisficble?	· · · ·		· · · · ·	· · · ·	· · ·	· · · ·	· ·	· · ·	· · ·	· · ·
	· · · ·		· · · · ·	· · ·	· ·	· · · ·	· ·	· ·	· ·	· ·
$X_1$ $X_2$ $X_3$ $C_1$ $C_2$ $C_3$		· · · ·	· · · · ·	· · ·	· ·	· · ·	· ·	· · ·	· ·	· ·
		· · · ·	· · · · ·	· · · ·	· ·	· · · ·	· ·	· ·	· ·	· ·
		· · · ·	· · · · ·	· · · ·	· ·	· · · ·	· · ·	· · ·	· ·	· ·
		· · · ·	· · · · ·	· · ·	· ·	· · · ·	· ·	· · ·	· ·	· ·
		· · · ·	· · · · ·	· · · ·	· ·	· · · ·	· ·	· ·	· ·	· ·
· · · · · · · · · · · · · · · · · · ·	· · · ·	· · · ·	· · · · ·	· · · ·	· ·	· · · ·	· ·	· ·	· ·	· ·

Announceme	<u>+s</u>	· · · ·
X HW5 D		· · · ·
* Prelim 2	Conflict Survey on Ed.	· · · ·
X Recitation	on Saturday	· · · ·
		· · · ·
	.       .	· · · ·
· · · · · · · · · · ·		· · · ·
	.       .	· · · ·
· · · · · · · · · · ·		· · · ·
· · · · · · · · · ·		· · · ·
	· · · · · · · · · · · · · · · · · · ·	

SAT.	Given a boolean formu	$r(\alpha, \phi)$	•
<del>.</del>	does there exist a	satisfying assignment?	•
			٠
			•
			•
			•
			٠
			•
			•
			•
			•
			٠
			•
			•
			•
			•
			•
			•
			•
			•

SAT. Given Joes	n a boolean there exist	formula a sat	isfying as	ssignment.
Observation.	Given on a	ssignment	$\overrightarrow{X} = \overrightarrow{\alpha}$	· · · · · · · · · · · · · · · · · · ·
	there is an	efficient	algorithm.	to verify
· · · · · · · · · · · · · ·	$(\overline{\alpha}) =$	OOR	$(\overline{o}, \overline{o}) = (1)$	· · · · · · · · · · ·
· · · · · · · · · · · · · · ·	· · · · · · · · · · · · ·	· · · · · · · ·	· · · · · · · · · ·	· · · · · · · · · · ·
· · · · · · · · · · · · · · ·	· · · · · · · · · · · · ·	· · · · · · · ·	· · · · · · · · · · ·	· · · · · · · · · · · ·
· · · · · · · · · · · · ·	· · · · · · · · · · · · ·		· · · · · · · · · ·	
· · · · · · · · · · · · · ·	· · · · · · · · · · · · ·	· · · · · · · ·	· · · · · · · · · ·	· · · · · · · · · · ·
· · · · · · · · · · · · · · ·	· · · · · · · · · · · · ·	· · · · · · · ·	· · · · · · · · · ·	· · · · · · · · · · ·

SAT. Given a boolean formula \$, does there exist a satisfying assignment?
Observation. Given an assignment $\vec{x} = \vec{a}$ there is an efficient algorithm to verify $\phi(\vec{a}) = 0$ or $\phi(\vec{a}) = 1$
For each clause $C = (l_1 \vee l_2 \vee \cdots \vee l_n) \in \phi$ . [ Assign literals $l_1 \in \mathbb{Z} \times i_1 = \pi \times i_2$ according to $\alpha_i$ if $C(\alpha) = 0$ , Return False
Return True

SAT. Given a boolean formula $\phi$ , does there exist a satisfying assignment?
Observation. Given an assignment $\bar{x} = \bar{a}$ there is an efficient algorithm to verify
Solving SAT $\phi(\bar{\alpha}) = 0$ or $\phi(\bar{\alpha}) = 1$
Brute Force SAT Algorithm
For all assignments à E [0000, 0001,, 1111]
Verify $(\phi, a)$ If $\phi(a) = 1$ : Return True
Return False
Ketwo False

SAT. Given a boolean formula $\phi$ , does there exist a satisfying assignment?
Observation. Given an assignment $\vec{x} = \vec{a}$ there is an efficient algorithm to verify
Solving SAT $\phi(\bar{\alpha}) = 0$ or $\phi(\bar{\alpha}) = 1$
Brute Force SAT Algorithm
For all assignments à E [0000, 0001,, 1111]
Verify $(\phi, \hat{a})$ If $\phi(\hat{a}) = 1$ : Return True -Set each literal in each clause -Determine if there is a 1 in each clause

SAT. Given a boolean formula $\phi$ , does there exist a satisfying assignment?
Observation. Given an assignment $\tilde{x} = \tilde{a}$ there is an efficient algorithm to verify
Solving SAT $\phi(\bar{\alpha}) = 0$ or $\phi(\bar{\alpha}) = 1$ = 2 <sup>h</sup>
Brute Force SAT Algorithm
For all assignments à E [0000, 0001,, 1111]
Verify $(\phi, \bar{\alpha})$ If $\phi(\bar{\alpha}) = 1$ : Return True -Set each literal in each clause Return False $\Rightarrow \Theta(2^n, poly(n, m))$ G(n, m)

•	· · ·	*		2 2 2	ر <del>و</del>		•	•	•		T	, 2 2	و م	~ <	2	L	2-Y	<	5	(–	Ś	•			•	•	•		) ) ,	2	ν	•	Po	الم	(رر	$\sim$ , '	m	) \	) .	•	* * *	+	in	مە	•		•	•	•
•	• •	•	•	0	•		•	*	0	•		• С	C	20	10,	י י ער ע	H	N	~	•		ſ	σι		•	0	, ,	۔ 1 س	J.I			S	Ą	<u>,</u>	- · ·.		0	*		0	•	0	•	•	•	0	•	•	•
•	• •	•	•	0	•	•	•	•	•	•		•	•	. `	J .			•	•	•	•			•	•	•	•	•	•	•	•	0	•	0	•	•	0	•	•	•	•	•		•	•	•	•	•	•
•		•	•									•						•															٠		٠									٠					
•		Ė	> >e	He	$\sim$	•	Å	(	S	D.	, v	. +	i (,	M	حە	•	? °		•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•
	· •	•	•		•	•		•		•		•	•	•	•	ſ		0	+	•		4	e -	F			•	•	•	•	•	•	•	•	•	•	0	•		•	•	•	•	•	•	•	9 0		•
•	• •	•	•	•	•	•	•	•	0	•		•	•	•	•	•		•	•		•	· .		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•		•	•	•	•	•	•	•	•	•		•	•	•	•	•		•	•	•	•	•		•	•	•	•	•	•	•	•	•	÷	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•		•	•	•	•	•	•	•	•	•		•	•	•	•			•	•	•	•			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•	•	•
•	• •	•	•	•	•	•	•	•	•	•		•	•		•	•		•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
										•		*			•	•		•			•			•			•			•	•		•	0		•	0	•	•			0		•	•		•		
•		•	•	•	•	•	•	•	•	÷		•	•	•	•	•		•	•	•	•			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•		•	•	•	•	•	•	•	•	•		•	•	•	•	•		•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•		•	•	•	•	•	•	•	•	•		•	•	•	•	•		•	•	•	•			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•		•	•	0	•		•	•	•	•		•	•	•	•	•		•	•	•	•	•		•	•	•	•	•	•	•	•	0	•	0	•	•	0	•	•	•	•	•	•	•	•		•	•	•
•		•	•	•	•	•	•	•	•	•		*	•		•			•	•	•	•			•	•	•	•		•	•	•		•	•	•	•	•	•	•	•	•	•			•	•	•	•	

Theorem T	here exists a	$O(2^n \operatorname{poly}(n,m))$ CNF-SAT	) - time	  
	ула ула на	.       .	  	.     .     .     .     .       .     .     .     .     .       .     .     .     .     .       .     .     .     .     .       .     .     .     .     .
De Hev Migsv	Not yet	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$		· · · · · ·
No known	$O_{1}\left(1,999,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1$	-ime algorithm	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	· · · · · · · ·
	$O_{1}\left(1:999\right) + B_{1}$	ime algorithm ?		$\begin{array}{cccccccccccccccccccccccccccccccccccc$

· ·		1 _0U	νF			;+	 	•		1 7 h	م کرم			•	• •	•	•		•	· ·	•	•	••••	•	• •		• •	· ·	•		• •	•	•	· ·	•	•
· · ·		•		ر مر	-e	ہ در د	, 1 , √ , √	, 1 Z	e_	•		)v.c	<u>, b</u>	Lei	 	کې د	•	60	ر ج ر	ed	•		, , , , , , ,	1	~. ~.~	ہ مر ر		Ēŀ	4 S	Ŷ		H	A R	D.	•	•
· ·	•	•	• •	• •	•	•	• •		•	•	• •	•	•	•	• •	•	•	•	•	• •	•		th	د میرد	 1 .	c C	NR	· ·	Ţ	0.7	•••	Sc	ہ∫` ۔ `	e	• .	•
	•	•	• •		•	•	• •	•	•	•	• •	•	•	•	• •	•	•	•	•	• •	•	•	• •	•	• •	•	• •		•	•	• •	•	•	• •	•	•
· ·	•	•	• •		•	•	• •	•	•		• •	•		а Ф	• •	•	•	•		• •	•	•	· ·	•	· ·	•	• •	• •	•	•	• •	•	•	• •	•	•
· ·	•	•	· ·	• •	•	•	• •		•	•	• •			•	• •	•	•	•	•	· ·	•	•	· ·	•	· ·		• •	• •	•	•	• •		•	· ·		•
• •	•	•	• •	•	•	•	• •	•	•	•	• •	•	•	•	• •	•	•	•	•	• •	•	•	· ·	•	• •	•	• •	• •	•	•	••••	•	•	• •	•	•
• •	•	•	• •	• •	•	•	• •	•	•	•	• •	•	•	•	• •	•	•	•	•	• •	•	•	· ·	•	• •	•	• •	• •	•	•	••••	•	•	• •	•	•
· ·	•	•	• •	• •	•	•	• •		•	•	• •	•	•	•	• •	•	•	•	•	••••	•	•	· ·	•	• •	•	• •	• •	•	•	••••	•	•	• •	•	•
	•	•		•	•	•	••••	•	•	•	• •	•	•	•	• •	•	•	•	•	• •	•	•	· ·	•		•	• •		•	•	••••	•	•	• •	•	•
• •		•	• •	- • •		•	• •	•	•		• •			•	• •			•		• •		•	• •			•				•	• •	•	•	0 0		
• •	•	•	• •	• •	•	•	• •	•	•	•	••••	•	•	•	• •	•	•	•	•	• •	•	•	· ·	•	· ·	•	• •	· ·	•	•	••••	•	•	• •	•	•
· ·	•	•	• •	• •	•	•	· ·	•	•	•	• •	•	•	•	• •	•	•	•	•	• •	•	•	· ·	•	· ·	•	• •	• •	•	•	· ·	•	•	• •	•	•

Comp	lexity The		· · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
	ategorize	problems be	ased on ho	~ EASY/HARD
	· · · · · · · · · ·		they	ave to solve
· · · · · · ·		· · · · · · · · · ·		· · · · · · · · · · · · · · · · ·
	Problems	that can	be solved	in polynomial time
		· · · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · ·			· · · · · · · · · ·	· · · · · · · · · · · · · · · ·
· · · · · · ·	· · · · · · · · · ·	· · · · · · · · · ·	· · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
				· · · · · · · · · · · · · · · · ·
· · · · · · ·		· · · · · · · · · ·	· · · · · · · · · ·	· · · · · · · · · · · · · · · · ·
		· · · · · · · · · · ·		· · · · · · · · · · · · · · · · · · ·
· · · · · · · ·	· · · · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·

Compl	exity The		· · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
	tegorize	problems b	ased on h	~ EASY/HARD
· · · · · · · ·	· · · · · · · · ·	· · · · · · · · · ·	they	ave to solve
· · · · · · · · · · · · · · · · · · ·	Problems	that can	be solved	in polynomial time.
$N = \frac{1}{2}$	Problems	that can	be verified	in polynomial time
· · · · · · · · ·		· · · · · · · · · · ·	· · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · ·		· · · · · · · · · ·		.       .
· · · · · · · · ·		.       .	.       .	

Complexity Theory
Categorize problems based on how EASY/HARD
they are to solve.
PE Problems that can be solved in polynomial time.
NP = Problems that can be verified in polynomial time.
P vs. NP. Does efficient verification imply efficient solving?

· · · · · · · · · · · · · ·		
		NPA a serie
	ST =DIST SHOULTEST-PATH	SAT
MAX BIP	MATCH	
	$\mathcal{F}_{LOW} = \mathcal{F}_{M} = \mathcal{F}_{M$	
	· · · · · · · · · · · · · · · · · · ·	
· · · · · · · · · · · · ·		· · · · · · · · · · · · · · · · · · ·
		· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · ·		· · · · · · · · · · · · · · · · · · ·

	MST EDIT-DIST MAX BIP. MATCH MAX FLOW	P SHOUTEST-PAT MINCUT	·       ·				
Reduction Suppose Q =	s as a m R veduces	Lasure of	havdnes via poly	- time ve	duction .		
· · · · · · · · · ·		· · · · · · · · ·	· · · · · · · ·	· · · · · ·		· · · · · · · ·	· · · ·

MST EDIFDIST SHOLTEST-PATH EDIFDIST SHOLTEST-PATH MAX BIP. MATCH MAX FLOW
Reductions as a measure of hardness
Suppose Q reduces to R via poly-time reduction. $Q \leq_p R$ If R is easy, so is Q.

MST EDIFDIST SHOLTEST-PATH MAX BIP. MATCH MAX FLOW	
Reductions as a measure of hardness Suppose Q reduces to R via poly-time reduction.	
If R is easy, so is Q $If Q is Hard, so is R$	· · · · · · · · · · · · · · · · · · ·

MST EDIT-DIST MAX BIP. MATCH MAX FLOW	CIRCUIT SAT SHOUTEST-PATTH FACTORING NEUT SUBSET SUM	N.P. SAT.	
	VERTEX-COVER		•
Reductions as a mea	sure of hardness		
Suppose Q reduces $f_{p}$ $Q \leq_{p} R$	If R is easy	r veduction, $r$ is a second secon	•
.       .	If Q is Hard,	$s_{s}$ $s_{s}$ $s_{s}$ $R_{s}$ $s_{s}$ $R_{s}$ $s_{s}$	•

P CIRCUITSAT MST SHOLTEST-PATT EDITOIST SAT FACTORING MAX BIP. MATCH MINCUT SUBSET MAXFLOW IND SET VERTEX-COVER Theorem. Every problem QENP reduces to SAT

MST EDITEDIST SHOWTEST-PATTH FACTORING SAT
MAX BIP. MINL SUBSET SUM MAX FLOW IND SET
Theorem. Every problem QENP reduces to SAT!
If any efficiently-verifiable problem is hand to solve, then SAT is hand to solve.

Complexity Theory
Categorize problems based on how EASY/HARD
they are to solve
P = Problems that can be solved in polynomial time.
NP = Problems that can be verified in polynomial time.
NP-Hard Z Q is NP-Hand if every problem in NP reduces to Q.

NP-Complete	
Q is NP-Complete if.	
$\times$ $Q \in NP$	· · · · · · · · · · · · · · · · · · ·
* Q is NP-Hand	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
The Cook-Levin Theorem	SAT is NP-Complete.
The Cook-Levin Theorem	SAT is NP-Complete
The Cook - Levin Theorem	SAT is NP-Complete.
The Cook-Levin Theorem	SAT is NP-Complete.
The Cook-Levin Theorem	SAT is NP-Complete.
The Cook - Levin Meorem	SAT is NP-Complete.