SATisfiability

Satisfiability (SAT)

Def $\phi(\vec{x}) \in \mathrm{SAT}$ if there is $\vec{b}$ such that $\phi(\vec{b}) = T$.
If $\vec{b}$ exists it is called a **Satisfying Assignment**.

Satisfiability (SAT)

Def $\phi(\vec{x}) \in \mathrm{SAT}$ if there is $\vec{b}$ such that $\phi(\vec{b}) = T$.
If $\vec{b}$ exists it is called a **Satisfying Assignment**.

$$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2) \wedge (\neg x_2 \vee \neg x_3) \in \mathrm{SAT}?$$

Satisfiability (SAT)

Def $\phi(\vec{x}) \in \mathrm{SAT}$ if there is $\vec{b}$ such that $\phi(\vec{b}) = T$.
If $\vec{b}$ exists it is called a **Satisfying Assignment**.

$$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2) \wedge (\neg x_2 \vee \neg x_3) \in \mathrm{SAT}?$$

Yes $x_1 = T$, $x_2 = F$, $x_3 = F$.

Satisfiability (SAT)

Def $\phi(\vec{x}) \in \mathrm{SAT}$ if there is $\vec{b}$ such that $\phi(\vec{b}) = T$.
If $\vec{b}$ exists it is called a **Satisfying Assignment**.

$$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2) \wedge (\neg x_2 \vee \neg x_3) \in \mathrm{SAT}?$$

Yes $x_1 = T$, $x_2 = F$, $x_3 = F$.

$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_2 \vee \neg x_3) \wedge x_2 \in \mathrm{SAT}?$$

Satisfiability (SAT)

Def $\phi(\vec{x}) \in \mathrm{SAT}$ if there is $\vec{b}$ such that $\phi(\vec{b}) = T$.
If $\vec{b}$ exists it is called a **Satisfying Assignment**.

$$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2) \wedge (\neg x_2 \vee \neg x_3) \in \mathrm{SAT?}$$

Yes $x_1 = T$, $x_2 = F$, $x_3 = F$.

$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_2 \vee \neg x_3) \wedge x_2 \in \mathrm{SAT?}$$

NO

Satisfiability (SAT)

Def $\phi(\vec{x}) \in \mathrm{SAT}$ if there is $\vec{b}$ such that $\phi(\vec{b}) = T$.
If $\vec{b}$ exists it is called a **Satisfying Assignment**.

$$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2) \wedge (\neg x_2 \vee \neg x_3) \in \mathrm{SAT?}$$

Yes $x_1 = T$, $x_2 = F$, $x_3 = F$.

$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_2 \vee \neg x_3) \wedge x_2 \in \mathrm{SAT?}$$

NO

If $x_1 = T$ then $x_3 = T$, $x_2 = F$. NO GOOD.

Satisfiability (SAT)

Def $\phi(\vec{x}) \in \mathrm{SAT}$ if there is $\vec{b}$ such that $\phi(\vec{b}) = T$.
If $\vec{b}$ exists it is called a **Satisfying Assignment**.

$$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2) \wedge (\neg x_2 \vee \neg x_3) \in \mathrm{SAT}?$$

Yes $x_1 = T$, $x_2 = F$, $x_3 = F$.

$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_2 \vee \neg x_3) \wedge x_2 \in \mathrm{SAT}?$$

NO

If $x_1 = T$ then $x_3 = T$, $x_2 = F$. NO GOOD.
If $x_2 = F$ then $x_2 = F$. NO GOOD.

Complexity of Satisfiability

SAT Problem Given ϕ, determine if $\phi \in \mathrm{SAT}$.

Complexity of Satisfiability

SAT Problem Given ϕ, determine if $\phi \in \mathrm{SAT}$.

One Approach Form Truth Table and see if any of the rows are T.
This is often called a **brute force search**.
What are the PROS and CONS of this approach?

Complexity of Satisfiability

SAT Problem Given ϕ, determine if $\phi \in \mathrm{SAT}$.

One Approach Form Truth Table and see if any of the rows are T.
This is often called a **brute force search**.
What are the PROS and CONS of this approach?

1. **PRO** Easy conceptually. Easy to code up.

Complexity of Satisfiability

SAT Problem Given ϕ, determine if $\phi \in \mathrm{SAT}$.

One Approach Form Truth Table and see if any of the rows are T.
This is often called a **brute force search**.
What are the PROS and CONS of this approach?

1. **PRO** Easy conceptually. Easy to code up.
2. **CON** Takes time roughly 2^n in the worst case.

Complexity of Satisfiability

SAT Problem Given ϕ, determine if $\phi \in \mathrm{SAT}$.

One Approach Form Truth Table and see if any of the rows are T.
This is often called a **brute force search**.
What are the PROS and CONS of this approach?

1. **PRO** Easy conceptually. Easy to code up.
2. **CON** Takes time roughly 2^n in the worst case.
3. **CAVEAT** Might do well on a formula that is in SAT since the algorithm can quit as soon as it finds a satisfying assignment.

On the next few slides discuss the following:

Complexity of Satisfiability

SAT Problem Given ϕ, determine if $\phi \in \mathrm{SAT}$.

One Approach Form Truth Table and see if any of the rows are T. This is often called a **brute force search**.
What are the PROS and CONS of this approach?

1. **PRO** Easy conceptually. Easy to code up.
2. **CON** Takes time roughly 2^n in the worst case.
3. **CAVEAT** Might do well on a formula that is in SAT since the algorithm can quit as soon as it finds a satisfying assignment.

On the next few slides discuss the following:

1. Is there a better algorithm?

Complexity of Satisfiability

SAT Problem Given ϕ, determine if $\phi \in \mathrm{SAT}$.

One Approach Form Truth Table and see if any of the rows are T. This is often called a **brute force search**.
What are the PROS and CONS of this approach?

1. **PRO** Easy conceptually. Easy to code up.
2. **CON** Takes time roughly 2^n in the worst case.
3. **CAVEAT** Might do well on a formula that is in SAT since the algorithm can quit as soon as it finds a satisfying assignment.

On the next few slides discuss the following:

1. Is there a better algorithm?
2. Is there a class of formulas for which there is a better algorithm?

Complexity of Satisfiability

SAT Problem Given ϕ, determine if $\phi \in \mathrm{SAT}$.

One Approach Form Truth Table and see if any of the rows are T.
This is often called a **brute force search**.
What are the PROS and CONS of this approach?

1. **PRO** Easy conceptually. Easy to code up.
2. **CON** Takes time roughly 2^n in the worst case.
3. **CAVEAT** Might do well on a formula that is in SAT since the algorithm can quit as soon as it finds a satisfying assignment.

On the next few slides discuss the following:

1. Is there a better algorithm?
2. Is there a class of formulas for which there is a better algorithm?
3. Is this problem interesting to people outside of Logic?

Is There a Better Algorithm?

Writing out the truth table takes roughly 2^n steps.

Is There a Better Algorithm?

Writing out the truth table takes roughly 2^n steps.

Is there a better algorithm. **Vote**

Is There a Better Algorithm?

Writing out the truth table takes roughly 2^n steps.
Is there a better algorithm. **Vote**

- YES
- NO
- UNKNOWN TO SCIENCE

Is There a Better Algorithm?

Writing out the truth table takes roughly 2^n steps.
Is there a better algorithm. **Vote**

- YES
- NO
- UNKNOWN TO SCIENCE

YES and **UNKNOWN TO SCIENCE**

Is There a Better Algorithm?

Writing out the truth table takes roughly 2^n steps.
Is there a better algorithm. **Vote**

- YES
- NO
- UNKNOWN TO SCIENCE

YES and **UNKNOWN TO SCIENCE**
YES If ϕ is in 3-CNF form (we'll define that later) then there exists a randomized 1.306^n algorithm.

Is There a Better Algorithm?

Writing out the truth table takes roughly 2^n steps.
Is there a better algorithm. **Vote**

- YES
- NO
- UNKNOWN TO SCIENCE

YES and **UNKNOWN TO SCIENCE**

YES If ϕ is in 3-CNF form (we'll define that later) then there exists a randomized 1.306^n algorithm.

UNKNOWN TO SCIENCE If there are no restrictions on the formula, then unknown if there is an algorithm better than $\sim 2^n$.

What is Better?

There are many algorithms that work in time α^n for some $1 < \alpha < 2$.

What is Better?

There are many algorithms that work in time α^n for some $1 < \alpha < 2$.

- These algorithms are very clever but are still **Brute Force Search with Tricks**.

What is Better?

There are many algorithms that work in time α^n for some $1 < \alpha < 2$.

- These algorithms are very clever but are still **Brute Force Search with Tricks**.
- We want to say **An Algorithm that is NOT brute force Search with Tricks**. How can we define that?

What is Better?

There are many algorithms that work in time α^n for some $1 < \alpha < 2$.

- These algorithms are very clever but are still **Brute Force Search with Tricks**.
- We want to say **An Algorithm that is NOT brute force Search with Tricks**. How can we define that?

Contrast:

What is Better?

There are many algorithms that work in time α^n for some $1 < \alpha < 2$.

- These algorithms are very clever but are still **Brute Force Search with Tricks**.
- We want to say **An Algorithm that is NOT brute force Search with Tricks**. How can we define that?

Contrast:

- There is an algorithm for SAT that takes $\sim (1.1)^n$.

What is Better?

There are many algorithms that work in time α^n for some $1 < \alpha < 2$.

- These algorithms are very clever but are still **Brute Force Search with Tricks**.
- We want to say **An Algorithm that is NOT brute force Search with Tricks**. How can we define that?

Contrast:

- There is an algorithm for SAT that takes $\sim (1.1)^n$.
- There is an algorithm for SAT that takes $\sim n^{100}$.

What is Better?

There are many algorithms that work in time α^n for some $1 < \alpha < 2$.

- These algorithms are very clever but are still **Brute Force Search with Tricks**.
- We want to say **An Algorithm that is NOT brute force Search with Tricks**. How can we define that?

Contrast:

- There is an algorithm for SAT that takes $\sim (1.1)^n$.
- There is an algorithm for SAT that takes $\sim n^{100}$.

In practice the $(1.1)^n$ algorithm is better.

What is Better?

There are many algorithms that work in time α^n for some $1 < \alpha < 2$.

- These algorithms are very clever but are still **Brute Force Search with Tricks**.
- We want to say **An Algorithm that is NOT brute force Search with Tricks**. How can we define that?

Contrast:

- There is an algorithm for SAT that takes $\sim (1.1)^n$.
- There is an algorithm for SAT that takes $\sim n^{100}$.

In practice the $(1.1)^n$ algorithm is better.

However, the n^{100} algorithm **is not doing brute force search!**

Polynomial Time

We now have our clean question:

Is SAT in Polynomial Time?

Polynomial Time

We now have our clean question:

Is SAT in Polynomial Time?

Question If SAT is in time n^{100} why do we care?

Polynomial Time

We now have our clean question:

Is SAT in Polynomial Time?

Question If SAT is in time n^{100} why do we care?

Answer If SAT is in time n^{100} then there is an algorithm that solves SAT that is not doing brute force search. It is doing **something clever**. That cleverness can likely be used to come up with a **much** better algorithm.

Polynomial Time

We now have our clean question:

Is SAT in Polynomial Time?

Question If SAT is in time n^{100} why do we care?

Answer If SAT is in time n^{100} then there is an algorithm that solves SAT that is not doing brute force search. It is doing **something clever**. That cleverness can likely be used to come up with a **much** better algorithm.

Notation We denote Polynomial Time by **P**.

Is There a Class of Formulas for Which SAT is in P?

We define several variants of SAT:

Is There a Class of Formulas for Which SAT is in P?

We define several variants of SAT:

1. SAT is the set of all formulas that are satisfiable.

Is There a Class of Formulas for Which SAT is in P?

We define several variants of SAT:

1. SAT is the set of all formulas that are satisfiable. That is, $\phi(\vec{x}) \in SAT$ if there exists a vector $\vec{b}$ such that $\phi(\vec{b}) = T$.

Is There a Class of Formulas for Which SAT is in P?

We define several variants of SAT:

1. SAT is the set of all formulas that are satisfiable. That is, $\phi(\vec{x}) \in SAT$ if there exists a vector $\vec{b}$ such that $\phi(\vec{b}) = T$.
2. CNFSAT is the set of all formulas in SAT of the form $C_1 \wedge \cdots \wedge C_m$ where each C_i is an $\vee$ of literals.

Is There a Class of Formulas for Which SAT is in P?

We define several variants of SAT:

1. SAT is the set of all formulas that are satisfiable. That is, $\phi(\vec{x}) \in SAT$ if there exists a vector $\vec{b}$ such that $\phi(\vec{b}) = T$.
2. CNFSAT is the set of all formulas in SAT of the form $C_1 \wedge \cdots \wedge C_m$ where each C_i is an $\vee$ of literals.
3. k-CNFSAT is the set of all formulas in SAT of the form $C_1 \wedge \cdots \wedge C_m$ where each C_i is an $\vee$ of exactly k literals.

Is There a Class of Formulas for Which SAT is in P?

We define several variants of SAT:

1. SAT is the set of all formulas that are satisfiable. That is, $\phi(\vec{x}) \in SAT$ if there exists a vector $\vec{b}$ such that $\phi(\vec{b}) = T$.
2. CNFSAT is the set of all formulas in SAT of the form $C_1 \wedge \cdots \wedge C_m$ where each C_i is an $\vee$ of literals.
3. k-CNFSAT is the set of all formulas in SAT of the form $C_1 \wedge \cdots \wedge C_m$ where each C_i is an $\vee$ of exactly k literals.
4. DNFSAT is the set of all formulas in SAT of the form $C_1 \vee \cdots \vee C_m$ where each C_i is an $\wedge$ of literals.

Is There a Class of Formulas for Which SAT is in P?

We define several variants of SAT:

1. SAT is the set of all formulas that are satisfiable. That is, $\phi(\vec{x}) \in SAT$ if there exists a vector $\vec{b}$ such that $\phi(\vec{b}) = T$.
2. CNFSAT is the set of all formulas in SAT of the form $C_1 \wedge \cdots \wedge C_m$ where each C_i is an $\vee$ of literals.
3. k-CNFSAT is the set of all formulas in SAT of the form $C_1 \wedge \cdots \wedge C_m$ where each C_i is an $\vee$ of exactly k literals.
4. DNFSAT is the set of all formulas in SAT of the form $C_1 \vee \cdots \vee C_m$ where each C_i is an $\wedge$ of literals.
5. k-DNFSAT is the set of all formulas in SAT of the form $C_1 \vee \cdots \vee C_m$ where each C_i is an $\wedge$ of exactly k literals.

2-CNFSAT is in P

2-CNFSAT is $C_1 \wedge \cdots \wedge C_m$ where each C_i is an $\vee$ of exactly 2 literals.

2-CNFSAT is in P

2-CNFSAT is $C_1 \wedge \cdots \wedge C_m$ where each C_i is an $\vee$ of exactly 2 literals.

2-CNFSAT is in P. Might be a HW. Intuition for now. Consider

$$(x \vee y).$$

2-CNFSAT is in P

2-CNFSAT is $C_1 \wedge \cdots \wedge C_m$ where each C_i is an $\vee$ of exactly 2 literals.

2-CNFSAT is in P. Might be a HW. Intuition for now. Consider

$$(x \vee y).$$

If x F then y T.

2-CNFSAT is in P

2-CNFSAT is $C_1 \wedge \cdots \wedge C_m$ where each C_i is an $\vee$ of exactly 2 literals.

2-CNFSAT is in P. Might be a HW. Intuition for now. Consider

$$(x \vee y).$$

If x F then y T.
More generally, with 2-CNFSAT a lot of values are forced.

2-CNFSAT is in P

2-CNFSAT is $C_1 \wedge \cdots \wedge C_m$ where each C_i is an $\vee$ of exactly 2 literals.

2-CNFSAT is in P. Might be a HW. Intuition for now. Consider

$$(x \vee y).$$

If x F then y T.
More generally, with 2-CNFSAT a lot of values are forced.

Usually called 2-SAT.

DNFSAT is in P

DNFSAT is the set of all formulas in SAT of the form $C_1 \vee \cdots \vee C_m$ where each C_i is an $\wedge$ of literals.

DNFSAT is in P

DNFSAT is the set of all formulas in SAT of the form $C_1 \vee \cdots \vee C_m$ where each C_i is an $\wedge$ of literals.

DNFSAT is in P.

DNFSAT is in P

DNFSAT is the set of all formulas in SAT of the form $C_1 \vee \cdots \vee C_m$ where each C_i is an $\wedge$ of literals.

DNFSAT is in P.

Example $(x_1 \wedge \neg x_2 \wedge x_3) \vee \cdots$
The $\cdots$ means you can put any thing you want there.
Without knowing anything else, this formula is satisfiable.
Set $x_1 = T$, $x_2 = F$, $x_3 = T$.

DNFSAT is in P

DNFSAT is the set of all formulas in SAT of the form $C_1 \vee \cdots \vee C_m$ where each C_i is an $\wedge$ of literals.

DNFSAT is in P.

Example $(x_1 \wedge \neg x_2 \wedge x_3) \vee \cdots$
The $\cdots$ means you can put any thing you want there.
Without knowing anything else, this formula is satisfiable.
Set $x_1 = T$, $x_2 = F$, $x_3 = T$.

More Generally

DNFSAT is in P

DNFSAT is the set of all formulas in SAT of the form $C_1 \vee \cdots \vee C_m$ where each C_i is an $\wedge$ of literals.

DNFSAT is in P.

Example $(x_1 \wedge \neg x_2 \wedge x_3) \vee \cdots$
The $\cdots$ means you can put any thing you want there.
Without knowing anything else, this formula is satisfiable.
Set $x_1 = T$, $x_2 = F$, $x_3 = T$.

More Generally Given $\phi = C_1 \vee \cdots C_m$ where each C_i is a $\wedge$ of literals,

DNFSAT is in P

DNFSAT is the set of all formulas in SAT of the form $C_1 \vee \cdots \vee C_m$ where each C_i is an $\wedge$ of literals.

DNFSAT is in P.

Example $(x_1 \wedge \neg x_2 \wedge x_3) \vee \cdots$
The $\cdots$ means you can put any thing you want there.
Without knowing anything else, this formula is satisfiable.
Set $x_1 = T$, $x_2 = F$, $x_3 = T$.

More Generally Given $\phi = C_1 \vee \cdots C_m$ where each C_i is a $\wedge$ of literals,

- If there is some C_i that does not have both a variable and its negation, then $\phi \in \mathrm{DNFSAT}$.

DNFSAT is in P

DNFSAT is the set of all formulas in SAT of the form $C_1 \vee \cdots \vee C_m$ where each C_i is an $\wedge$ of literals.

DNFSAT is in P.

Example $(x_1 \wedge \neg x_2 \wedge x_3) \vee \cdots$
The $\cdots$ means you can put any thing you want there.
Without knowing anything else, this formula is satisfiable.
Set $x_1 = T$, $x_2 = F$, $x_3 = T$.

More Generally Given $\phi = C_1 \vee \cdots C_m$ where each C_i is a $\wedge$ of literals,

- If there is some C_i that does not have both a variable and its negation, then $\phi \in \mathrm{DNFSAT}$.
- Otherwise $\phi \notin \mathrm{DNFSAT}$.

Is 3-CNFSAT in P?

Is 3-CNFSAT in P? Vote:

Is 3-CNFSAT in P?

Is 3-CNFSAT in P? Vote:

- YES, and this is known (though perhaps complicated). Maybe it uses Ramsey Theory so I will be teaching it in my other class.

Is 3-CNFSAT in P?

Is 3-CNFSAT in P? Vote:

- YES, and this is known (though perhaps complicated). Maybe it uses Ramsey Theory so I will be teaching it in my other class.
- NO, and this is known, and the proof is difficult (proving things can't be done is usually hard). Maybe it uses Ramsey Theory so I will be teaching it in my other class.

Is 3-CNFSAT in P?

Is 3-CNFSAT in P? Vote:

- YES, and this is known (though perhaps complicated). Maybe it uses Ramsey Theory so I will be teaching it in my other class.
- NO, and this is known, and the proof is difficult (proving things can't be done is usually hard). Maybe it uses Ramsey Theory so I will be teaching it in my other class.
- UNKNOWN TO SCIENCE.

Is 3-CNFSAT in P?

Is 3-CNFSAT in P? Vote:

- YES, and this is known (though perhaps complicated). Maybe it uses Ramsey Theory so I will be teaching it in my other class.
- NO, and this is known, and the proof is difficult (proving things can't be done is usually hard). Maybe it uses Ramsey Theory so I will be teaching it in my other class.
- UNKNOWN TO SCIENCE.

UNKNOWN TO SCIENCE The $(1.306)^n$ algorithm is the best algorithm we know.

Is 3-CNFSAT in P?

Is 3-CNFSAT in P? Vote:

- YES, and this is known (though perhaps complicated). Maybe it uses Ramsey Theory so I will be teaching it in my other class.
- NO, and this is known, and the proof is difficult (proving things can't be done is usually hard). Maybe it uses Ramsey Theory so I will be teaching it in my other class.
- UNKNOWN TO SCIENCE.

UNKNOWN TO SCIENCE The $(1.306)^n$ algorithm is the best algorithm we know.

What Lower Bounds Are Known It is known that 3-CNFSAT cannot be done in $n^{1.8}$ time and log-space.

Is 3-CNFSAT in P?

Is 3-CNFSAT in P? Vote:

- YES, and this is known (though perhaps complicated). Maybe it uses Ramsey Theory so I will be teaching it in my other class.
- NO, and this is known, and the proof is difficult (proving things can't be done is usually hard). Maybe it uses Ramsey Theory so I will be teaching it in my other class.
- UNKNOWN TO SCIENCE.

UNKNOWN TO SCIENCE The $(1.306)^n$ algorithm is the best algorithm we know.

What Lower Bounds Are Known It is known that 3-CNFSAT cannot be done in $n^{1.8}$ time and log-space.

How Long Has It Been Open For? First posed in 1971, though see next slide.

Is this problem interesting?

Consider the following problems:

Is this problem interesting?

Consider the following problems:

1. **Traveling Salesperson Problem (TSP)** Given *n* cities and how much it costs to go from any city to an city, determine cheapest way to visit all cities, Studied since the 1930's.

Is this problem interesting?

Consider the following problems:

1. **Traveling Salesperson Problem (TSP)** Given n cities and how much it costs to go from any city to an city, determine cheapest way to visit all cities, Studied since the 1930's.
2. **Scheduling** Given n rooms and when the free, and given m people who are requesting them for certain timeslots, can you accommodates all of them? Studied since the 1880's.

Is this problem interesting?

Consider the following problems:

1. **Traveling Salesperson Problem (TSP)** Given n cities and how much it costs to go from any city to an city, determine cheapest way to visit all cities, Studied since the 1930's.
2. **Scheduling** Given n rooms and when the free, and given m people who are requesting them for certain timeslots, can you accommodates all of them? Studied since the 1880's.

The following is known:
(3-SAT is in P) $\leftrightarrow$ (TSP is in P) $\leftrightarrow$ (SCHED is in P).

Is this problem interesting?

Consider the following problems:

1. **Traveling Salesperson Problem (TSP)** Given n cities and how much it costs to go from any city to an city, determine cheapest way to visit all cities, Studied since the 1930's.
2. **Scheduling** Given n rooms and when the free, and given m people who are requesting them for certain timeslots, can you accommodates all of them? Studied since the 1880's.

The following is known:
(3-SAT is in P) $\leftrightarrow$ (TSP is in P) $\leftrightarrow$ (SCHED is in P).
There are **thousands** of problems are equiv to SAT. Hence:

Is this problem interesting?

Consider the following problems:

1. **Traveling Salesperson Problem (TSP)** Given n cities and how much it costs to go from any city to an city, determine cheapest way to visit all cities, Studied since the 1930's.
2. **Scheduling** Given n rooms and when the free, and given m people who are requesting them for certain timeslots, can you accommodates all of them? Studied since the 1880's.

The following is known:
(3-SAT is in P) $\leftrightarrow$ (TSP is in P) $\leftrightarrow$ (SCHED is in P).
There are **thousands** of problems are equiv to SAT. Hence:

- The complexity of 3-SAT is **important** since it relates to the complexity of many other problems.

Is this problem interesting?

Consider the following problems:

1. **Traveling Salesperson Problem (TSP)** Given n cities and how much it costs to go from any city to an city, determine cheapest way to visit all cities, Studied since the 1930's.
2. **Scheduling** Given n rooms and when the free, and given m people who are requesting them for certain timeslots, can you accommodates all of them? Studied since the 1880's.

The following is known:
(3-SAT is in P) $\leftrightarrow$ (TSP is in P) $\leftrightarrow$ (SCHED is in P).
There are **thousands** of problems are equiv to SAT. Hence:

- The complexity of 3-SAT is **important** since it relates to the complexity of many other problems.
- Many of the problems 3-SAT is equivalent to have been worked on for 90 or more years; hence, it is unlikely they are in P. Hence it is unlikely that 3-SAT is in P.

Proper Terminology and What Do People In the Know Think?

The problems SAT, TSP, and SCHED are three examples of problems in NP, which we are not going to define.

Proper Terminology and What Do People In the Know Think?

The problems SAT, TSP, and SCHED are three examples of problems in NP, which we are not going to define.

The question of SAT in P is often phrased as **Does P = NP?**

Proper Terminology and What Do People In the Know Think?

The problems SAT, TSP, and SCHED are three examples of problems in NP, which we are not going to define.

The question of SAT in P is often phrased as **Does P = NP?**

What does the Theory community think? Someone actually did a poll and discovered that 88% of the theorists polled think P≠NP (so $\mathrm{SAT} \notin$P).
If you want to see the poll, here is the link:

Proper Terminology and What Do People In the Know Think?

The problems SAT, TSP, and SCHED are three examples of problems in NP, which we are not going to define.

The question of SAT in P is often phrased as **Does P = NP?**

What does the Theory community think? Someone actually did a poll and discovered that 88% of the theorists polled think P$\neq$NP (so $\mathrm{SAT} \notin$P).
If you want to see the poll, here is the link:
`http://www.cs.umd.edu/~gasarch/papers/poll3.pdf`

Its not all Bad News I

Scenario Your boss wants you to solve the TSP problem. You know that finding the **optimal** solution is likely not easy to do. So you know to look for an **approximation**. Perhaps something that is at worst twice optimal.

Its not all Bad News I

Scenario Your boss wants you to solve the TSP problem. You know that finding the **optimal** solution is likely not easy to do. So you know to look for an **approximation**. Perhaps something that is at worst twice optimal.

More generally, if you now a problem is equivalent to SAT then you know that you should not look for an optimal poly time solutions. There are many other options to try.

Its not all Bad News II

In the year 2000 the Clay Math Institute set forth 7 mathematics problems that, if solved, they will give the solver $1,000,000.

Its not all Bad News II

In the year 2000 the Clay Math Institute set forth 7 mathematics problems that, if solved, they will give the solver $1,000,000.

Resolving P vs NP is one of them.

Its not all Bad News II

In the year 2000 the Clay Math Institute set forth 7 mathematics problems that, if solved, they will give the solver $1,000,000.

Resolving P vs NP is one of them. Go to it!

Its not all Bad News II

In the year 2000 the Clay Math Institute set forth 7 mathematics problems that, if solved, they will give the solver $1,000,000.

Resolving P vs NP is one of them. Go to it!

Warning My 4-year old great nephew Jase is already working on it. On his own he wrote down on a paper plate:

Its not all Bad News II

In the year 2000 the Clay Math Institute set forth 7 mathematics problems that, if solved, they will give the solver \$1,000,000.

Resolving P vs NP is one of them. Go to it!

Warning My 4-year old great nephew Jase is already working on it. On his own he wrote down on a paper plate:
$2 + 2 = 4$
$4 + 4 = 8$
$8 + 8 = 16$
$16 + 16 = 32$
$32 + 32 = 64$
$64 + 64 = 128$
$128 + 128 = 256$

Its not all Bad News II

In the year 2000 the Clay Math Institute set forth 7 mathematics problems that, if solved, they will give the solver $1,000,000.

Resolving P vs NP is one of them. Go to it!

Warning My 4-year old great nephew Jase is already working on it. On his own he wrote down on a paper plate:
$2+2=4$
$4+4=8$
$8+8=16$
$16+16=32$
$32+32=64$
$64+64=128$
$128+128=256$
He then ran out of room; however, his grandmother (my wife's sister) tells me he can go all the way to 2048.