Plan

* Beating Brute Force for 3SAT

* Announcements

* Orthogonal Vectors solves CNF-SAT
**CNF-SAT**

Given a CNF $\varphi = C_1 \land C_2 \land \ldots \land C_m$

Does there exist $\overline{a} \in \{0,1\}^n$ s.t. $\varphi(\overline{a}) = 1$?

$C_i = (d_i \lor d_{i_2} \lor d_{i_3} \lor \ldots \lor d_{i_n})$

**3SAT**

CNF-SAT where each clause has at most 3 literals
Brute Force SAT (4)

For each \( a \in \{0,1\}^n \). // \( 2^n \) possible assignments
Evaluate \( \Phi(a) \). // poly(n,m)
if \( \Phi(a) = 1 \) \( \rightarrow \) Return

Return \( X \)

Running Time? \( \rightarrow \) \( 2^n \cdot \text{poly}(n,m) \)
**Brute Force SAT** \( (\Psi) \)

For each \( a \in \{0,1\}^n \) \( // \ 2^n \) possible assignments

Evaluate \( \Psi(a) \) \( // \) poly\((n,m)\)

if \( \Psi(a) = 1 \) \( \rightarrow \) Return \( \checkmark \)

Return \( \times \)

Running Time? \( \rightarrow \) \( 2^n \cdot \text{poly}(n,m) \)

Can we do better?

\( \Rightarrow \) e.g. for 3SAT?
Given $\mathcal{F}$, $l_i \in \mathcal{F} \xi_i, \neg \xi_i \in \mathcal{F}$

Let $\mathcal{F} \vert_{l_i=b}$ be the simplification of $\mathcal{F}$ after setting all occurrences of $\xi_i$ consistent with $l_i = b$. 
Given $\mathcal{Y}$, $\mathcal{Y} \in \mathcal{F}_{X_i, \neg X_i}$. Let $\mathcal{Y} \mid_{l_i=b}$ be the simplification of $\mathcal{Y}$ after setting all occurrences of $X_i$ consistent with $l_i = b$. Then

$$\mathcal{Y} = (x_i \lor \neg x_3) \land (\neg x_1 \lor x_5) \land (x_2 \lor x_4 \lor x_5)$$

$$\mathcal{Y} \mid_{\neg x_i = 1} = (x_i \lor \neg x_3) \land (\neg x_1 \lor x_5) \land (x_2 \lor x_4 \lor x_5)$$
Given $\phi$, let $l_i \in \exists x_i, \neg x_i \Sigma$

Let $\phi \mid_{l_i=b}$ be the simplification of $\phi$ after setting all occurrences of $x_i$ consistent w/ $l_i = b$.

$$\phi = (x_i \lor \neg x_3) \land (\neg x_i \lor x_5) \land (x_2 \lor x_4 \lor x_5)$$

$$\phi \mid_{\neg x_i = 1} = (x_i \lor \neg x_3) \land (\neg x_i \lor x_5) \land (x_2 \lor x_4 \lor x_5)$$
Branch $3SAT(\varphi)$ (Monien-Speckmeyer '86)

if $\varphi$ is a 2-CNF
   solve $2SAT(\varphi)$ in polynomial time.
else, find some clause $C = (l_1 \lor l_2 \lor l_3)$

Return \[ \left( \text{Branch } 3SAT \left( \varphi \mid l_1 = 1 \right) \right. \]
\[ \left. \lor \text{Branch } 3SAT \left( \varphi \mid l_1 = 0, l_2 = 1 \right) \right) \]
\[ \lor \text{Branch } 3SAT \left( \varphi \mid l_1 = 0, l_2 = 0, l_3 = 1 \right) \]

Correctness

At least 1 of $l_1$, $l_2$, $l_3$ must be set to 1.
Branch 3SAT ($\Psi$) Makes 3 Recursive calls

Branch 3SAT ($\Psi | l_1 = 1$)

Branch 3SAT ($\Psi | l_1 = 0, l_2 = 1$)

Branch 3SAT ($\Psi | l_1 = 0, l_2 = 0, l_3 = 1$)

Running Time?

For n-variable 3-CNFs

$$T(n) \leq T(n-1) + T(n-2) + T(n-3) + \text{poly}(n)$$
**Branch 3SAT** (\(\Psi\))

Makes 3 Recursive calls

\[
\begin{align*}
&\text{Branch 3SAT } (\Psi | l_1 = 1) \\
&\text{Branch 3SAT } (\Psi | l_1 = 0, l_2 = 1) \\
&\text{Branch 3SAT } (\Psi | l_1 = 0, l_2 = 0, l_3 = 1)
\end{align*}
\]

**Running Time?**

For \(n\)-variable 3-CNFs

\[
T(n) \leq T(n-1) + T(n-2) + T(n-3) + \text{poly}(n)
\]

\[
\Rightarrow \\
T(n) \leq 1.833^n
\]
Announcements

* Prelim Review, April 9, 7-9pm Gates 401

* Prelim #2, April 11, 7:30pm

  → Room assignments announced after break
  → Covering
    - Divide & Conquer
    - Flow
    - NP - Completeness

* Have a great spring break!
Orthogonal Vectors Problem (OV)

Given two lists $A, B$ each of $N$ vectors over $\mathbb{R}_0^m$

Does there exist

\[ 1 \leq i, j \leq N \text{ s.t. } A_i \text{ and } B_j \text{ are orthogonal?} \]

\[ A_i \cdot B_j = \sum_{u=1}^{m} A_{iu} \cdot B_{ju} = 0 \]
Orthogonal Vectors Problem (OV)

Given: Two lists $A, B$ each of $N$ vectors over $\mathbb{F}_0,1^m$

Does there exist $1 \leq i,j \leq N$ s.t. $A_i$ and $B_j$ are orthogonal?

Naive OV:

For $i = 1 \ldots N$

For $j = 1 \ldots N$

Test if $A_i \cdot B_j = 0$

Running Time $N^2 \cdot m$

$A_i \cdot B_j = \sum_{k=1}^{m} A_{ik} \cdot B_{jk} = 0$
Theorem. (Due to Ryan Williams, former 4820 student!)

If there exists an $N^{1.9}$ time algorithm for $OV$, then there exists a $1.94^n$ time algorithm for CNF-SAT.
Theorem. (Due to Ryan Williams, former 4820 student!)

If there exists an $N^{1.9}$ time algorithm for $OV$, then there exists a $1.94^n$ time algorithm for $CNF$-$SAT$.

This would be a MAJOR breakthrough in Algorithms & Complexity Theory.
Idea. Reduce CNF-SAT \rightarrow OV.

Exponential-time reduction

Given \( \Phi = C_1 \land C_2 \land \ldots \land C_m \)

*write down*

\[ A \quad \text{N} = 2^{n/2} \quad \text{m} \]

\[ B \quad \text{N} = 2^{n/2} \quad \text{m} \]

based on "partial assignments"
Partial Assignments

* Consider splitting the variables in half

\[ X_1, X_2, \ldots, X_{n/2} \quad | \quad X_{n/2+1}, X_{n/2+2}, \ldots, X_n \]
Partial Assignments

* Consider splitting the variables in half

\[ x_1, x_2, \ldots, x_{n/2}, x_{n/2+1}, x_{n/2+2}, \ldots, x_n \]

* For \( i \in \{0, 1, \ldots, n/2 \}, j \in \{0, 1, \ldots, n/2 \} \)

\[ (x_1, x_2, \ldots, x_{n/2}, x_{n/2+1}, x_{n/2+2}, \ldots, x_n) \rightarrow (i, j) \]

is an assignment to \( \bar{x} \)

and \( i, j \) are partial assignments.
CNF-SAT via OV.

For each \( i \in \{0, 1\}^{n/2} \)

\[
A_i \leftarrow \text{Partial Assignment Gadget} \left( x_1, \ldots, x_{n/2}, i \right)
\]

For each \( j \in \{0, 1\}^{n/2} \)

\[
B_j \leftarrow \text{Partial Assignment Gadget} \left( x_{n/2+1}, \ldots, x_n, j \right)
\]

Return \( OV(A, B) \)
Vectors indexed by partial assignments

Each index $i \in \mathbb{0,1,2}^{n/2}$ corresponds to an assignment to $X_1, X_2, \ldots, X_{n/2} \leftarrow i$.

Each $j \in \mathbb{0,1,2}^{n/2}$ corresponds to assignment $X_{n/2+1}, X_{n/2+2}, \ldots, X_n \leftarrow j$. 
Vector coordinates determined by satisfying clauses

\[ \psi = C_1 \land C_2 \land \ldots \land C_m \]

\[ A_i = \begin{cases} 0 & \text{if } x_1, x_2, \ldots, x_{n/2} \leftrightarrow i, \\ 1 & \text{satisfies clause } C_k, \\ \text{otherwise} & \end{cases} \]

\[ C_k = \left( x_2 \lor x_9 \lor x_{n/2} \lor \neg x_{n-10} \lor \ldots \lor \neg x_7 \right) \]
Vector coordinates determined by satisfying clauses

\[ y = C_1 \land C_2 \land \ldots \land C_m \]

\[ B_{ij} = \begin{cases} 0 & \text{if } X_{i_{\text{start}}} \neq X_{i_{\text{end}}} \ldots \neq X_n \leftarrow j \\ 1 & \text{satisfies clause } C_k \\ 0 & \text{otherwise} \end{cases} \]

\[ C_k = (x_2 \lor x_3 \lor x_n \lor \neg x_{n-10} \lor \ldots \lor \neg x_7) \]
Claim.

\[ A_{ij} \cdot B_{ijn} = 0 \] if and only if

the assignment

\((x_1, x_2, \ldots, x_{n/2}, x_{n/2+1}, x_{n/2+2}, \ldots, x_n) \leftarrow (i, j)\)

satisfies the clause \(C_k\)
Claim.
\[ A_i u \cdot B_j u = 0 \quad \text{if and only if} \]
the assignment
\[ (x_1, x_2, \ldots, x_{n/2}, x_{n/2+1}, x_{n/2+2}, \ldots, x_n) \leftrightarrow (i, j) \]
satisfies the clause \( C_k \).

Corollary. There exists orthogonal \( A_i \) and \( B_j \)
if and only if \( \phi \) is satisfiable.
Orthogonal Vectors Problem (OV) Solves CNF-SAT

Reduction

\[ 2 \times 2^{n/2} \cdot \text{poly}(n, m) \]

+ \( T_{OV}(2^{n/2}) \)

Suppose \( T_{OV}(N) = N^{1.9} \),

\[ \Rightarrow \text{CNF-SAT: } (2^{n/2})^{1.9} \leq 1.94^n \]
What did we show?

* New algorithmic approach for solving CNF-SAT. If CNF-SAT requires $\sim 2^n$ time, then OV requires $\sim N^2$ time.

* Hardness for polynomial-time.
What did we show?

* New algorithmic approach for solving CNF-SAT. 

L, we only need to improve OV.

* Hardness for polynomial-time.

If CNF-SAT requires \(2^n\) time, then OV requires \(N^2\) time.

\[\text{Theorem.} \] If CNF-SAT requires \(2^n\) time, then Edit Distance requires \(\tilde{O}(n^2)\) time.