Lecture 17

Outline:
- 3-colorability & Friends
- Subset Sum & Friends
- Hamiltonian Path & Friends

**Def**
A k-coloring of a graph $G$ is a partition of $V(G)$ into $\leq k$ disjoint sets s.t. $\forall (u,v) \in E(G)$, $u$ and $v$ belong to different sets.

$k$-Colorability = $\{G \mid G$ has a $k$-coloring$\}$

**Thm** 3-colorability is NP-complete.

**Proof:**
1. 3-colorability is in NP (exercise)
2. 3-colorability is NP-Hard

Roadmap:
- Reduction from 3-SAT
- Runtime Analysis
- Correctness analysis

- Idea: make a graph that can be 3-colored if $\Phi$ is satisfiable.
  - 2 gadgets correspond to truth values
  - 3 colors correspond to $T$ (true), $F$ (false), $N$ (neither)
how do you make sure no literal ever gets assigned to N?

have a special triangle (palette)

here is a gadget for the truth table of Or.

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_1 \lor X_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

This can be colored T iff (at least one variable is colored T)

e.g.
You can represent each clause by combining 2 or-gadgets. e.g. 

\((X_1 \lor \overline{X}_2 \lor X_3)\) corresponds to:

![Diagram](image)

As you can see, this means 5 vertices and 10 edges get added per clause. I've labelled the vertices \(V_1 \ldots V_5\)

So, in the end, the graph looks something like this:

For \(\varphi = \varphi_1 \land \varphi_2 \land \ldots \land \varphi_m\) (setting \(\varphi_1 = (X_1 \lor \overline{X}_2 \lor X_3)\))

![Diagram](image)

Gadgets for \(\varphi_1\)

Gadgets (and edges) for \(\varphi_2 \ldots \varphi_m\)
Formal Reduction:

On input a 3-CNF $\varphi = \varphi_1 \land \varphi_2 \land \ldots \land \varphi_m$
over variables $x_1, \ldots, x_n$

Construct $G = (V, E)$ as follows:

$$V = \{T, F, N^3\} \cup \bigcup_{i=1}^{n} \{x_i\} \cup \{\overline{x}_i\} \cup \bigcup_{j=1}^{m} \{V_{ij}, V_{2j}, V_{3j}, V_{4j}, V_{5j}\}$$

- $5$ gadget vertices for each clause
- $2$ new vertices for each literal
- the palette

$$E = \{(T,F),(T,N),(N,F)\} \leftarrow \text{palette edges}$$
$$\cup \bigcup_{i=1}^{n} \{(x_i, \overline{x}_i), (x_i, N), (\overline{x}_i, N)\} \leftarrow \text{forces one of } \{x_i, \overline{x}_i\} \text{ to be } T, \text{ one other one false}$$
$$\cup \bigcup_{j=1}^{m} \{(y_{ij}, V_{ij}), (y_{2j}, V_{2j}), (V_{ij}, V_{2j}), (V_{ij}, V_{3j}), (V_{2j}, V_{3j}), (V_{3j}, V_{4j}), (V_{4j}, V_{5j}), (V_{4j}, T), (V_{5j}, T)\}$$

(where $\varphi_j = y_{ij} \lor y_{2j} \lor y_{3j}$)

the $5$ gadget edges for each clause

Run-time: $|V| = 3 + 2n + 5m$ (n = # literals, m = # clauses)

= linear in input size, and can be built
in $O(|\varphi|)$ time.

$|E| = 3 + 3n + 10m$

can also be built in time $O(|\varphi|)$ time.
Correctness:

\[ \phi \text{ is satisfiable } \Rightarrow G \text{ can be 3-colored} \]

We've already argued this...

Name your 3 colors \( t, f, \bar{0} \)

for a satisfying assignment of \( x_1, \ldots, x_n \) to \( T/F \)

- if \( x_i \) is true color \( x_i \) \( t \)
- color \( \bar{x}_i \) \( f \)

- if \( x_i \) is false color \( x_i \) \( f \)

The gadgets allow you to color everything else.

\[ \phi \text{ is not satisfiable } \Rightarrow G \text{ cannot be 3-colored} \]

Let \( a, \ldots, a_n \) be an assignment of truth values to the literals \( x_1, \ldots, x_n \)

Since \( \phi \) cannot be satisfied, there must be some clause \( \phi_j \) that is not satisfied with \( a_1, \ldots, a_n \)

\[ \phi_j = (y_{ij} \lor y_{2j} \lor y_{3j}) \]

Since \( \phi_j \) is not satisfied, they must all be colored \( f \).

But then there is no way to color \( v_{ij}, \ldots, v_{3j} \)
friends of 3-colorability:

for \( k \geq 3 \) it is fairly straightforward to show that K-colorability is NP complete (via reduction from 3-colorability).

HINT: Do this on your HW.
also, scheduling (from 1st class)

Note: why do we sometimes talk about languages, and sometimes talk about problems?
Given a language \( L \), the corresponding decision problem is to decide if an instance is a yes instance or no instance.

ex. Language: 3SAT

Decision Problem: given \( \Phi \) accept if \( \Phi \in 3SAT \)
reject otherwise

Instance: any string

Yes-instance: \( \Phi \) s.t. \( \Phi \in 3SAT \)
No-instance: \( \Phi \) s.t. \( \Phi \notin 3SAT \)

thinking about things in terms of instances can help:
e.g. to reduce from 3SAT to 3-colorability

show yes instances map to yes instances
no instances map to no instances
\[ \text{SUBSET-SUM} = \{ \langle S, t \rangle \mid S \text{ is a set of integers, written in decimal}, \] 
\[ t \text{ is an integer, written in decimal}, \] 
\[ \exists s' \subseteq S \quad \text{s.t.} \quad \sum_{x \in s'} x = t \} \]

\textbf{Thm.} \quad \text{SUBSET-SUM is NP-complete}

1. \quad \text{SUBSET-SUM} \in \text{NP} \quad \text{(exercise)}
2. \quad \text{NP-hardness}

We will reduce from 3SAT.

Given \( \varphi \): 3CNF with \( n \) vars, \( m \) clauses.

Make a table: each row will correspond to a \( \# \) in your set. You will be forced to select rows to reach your target.

<table>
<thead>
<tr>
<th>Variables ( x_1, x_2, \ldots, x_n )</th>
<th>Clauses ( \varphi_1, \varphi_2, \varphi_3, \ldots, \varphi_m )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_1 ) 1 0 0 0 ( \bar{1} )</td>
<td>( \varphi_1 ) ( 0 ) ( 0 ) ( 0 )</td>
</tr>
<tr>
<td>( f_1 ) 1 0 0 0 ( 0 )</td>
<td>( \varphi_2 ) ( 0 ) ( 0 ) ( 1 )</td>
</tr>
<tr>
<td>( t_2 ) 0 1 0 0 ( 0 )</td>
<td>( \varphi_3 ) ( 0 ) ( 1 ) ( 0 )</td>
</tr>
<tr>
<td>( f_2 ) 0 1 0 0 ( 1 )</td>
<td>( \varphi_{\frac{m}{2}} ) ( 0 ) ( 1 ) ( 0 )</td>
</tr>
<tr>
<td>( t_n ) 0 0 0 0 ( 0 )</td>
<td>( \varphi_m ) ( 0 ) ( 0 ) ( 1 )</td>
</tr>
<tr>
<td>( f_n ) 0 0 0 0 ( 1 )</td>
<td>( \varphi_{n+1} ) ( 0 ) ( 0 ) ( 1 )</td>
</tr>
</tbody>
</table>

Suppose \( \varphi_1 = \)
\[ x_1 \lor \overline{x_2} \lor x_3 \]

Select \( t_i \) if we set \( x_i \) to be \( T \).
Select \( f_i \) if we set \( x_i \) to be \( F \).

Exactly 1 of \( t_i \) or \( f_i \) must be selected.

A 1 for \( t_i \) in position \( j \) means \( x_i \) appears in \( \varphi_j \).
A 1 for \( f_i \) in position \( j \) means \( \overline{x_i} \) appears in \( \varphi_j \).
but now, what do we set our target to in the \( \phi_j \) col's? Could be 1, 2, or 3! Solution add slack variable rows.

\[
\begin{array}{cccccccc}
X_1 & X_2 & X_3 & \ldots & X_i & \ldots & X_n & \phi_1 & \phi_2 & \ldots & \phi_j & \ldots & \phi_m \\
+1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
f_1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
f_2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
f_3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
f_i & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
f_m & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\psi_1}{\psi_i} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\psi_2}{\psi_i} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\psi_3}{\psi_i} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\psi_m}{\psi_i} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\text{target} & 1 & 1 & 1 & \ldots & 1 & 3 & 3 & \ldots & 3 & \frac{\phi_1}{\phi_i} & = x_1 \lor \overline{x_2} \lor x_3
\end{array}
\]

more formally: construct \( S \) as follows:

- Each integer in \( S \) has \( n + m \) decimal digits.
- \( S = \{ t_i \}_{i=1}^{\infty} \cup \{ f_i \}_{i=1}^{\infty} \cup \{ \phi_j \}_{j=1}^{\infty} \cup \{ \psi_j \}_{j=1}^{\infty} \)

where \( t_i \) has 1 in position \( x_i \) and in each col \( \phi_j \) that contains \( x_i \);
- \( f_i \) has 1 in position \( x_i \) and in each col \( \phi_j \) that contains \( x_i \);
- \( \psi_j = \phi_i \) has 1 in position \( \phi_i \); 0's everywhere else

\[
\frac{t}{t} = \frac{1}{1} \frac{3}{3} \ldots \frac{3}{3}
\]