**EUGENIAN PATH**

- Traverse graph edges so as to "visit" each edge once.

**HAMILTONIAN PATH**

- Traverse graph edges so as to "visit" each edge once, and return to the start.

**DFS Algorithm**

- \( O(V + E) \)

**Cuscin a path**

- \( O(V + E) \)

**Verify**

**NP-Complete**

- **NP-hard**
- \( N \) is prime

**P \( \neq \) NP**

- \( P \) \( \subseteq \) \( \text{SAT} \)
- \( \text{RFP} \) \( \subseteq \) any problem in \( \text{NP} \)
- \( \text{RFP} \) \( \subseteq \) any problem in \( \text{NP} \)

**Length**

- \( n \) is prime

\( \Theta(n) \) bits

**Satisifiability**

- \( \text{SAT} \)
- \( \text{SAT} \) \( \subseteq \) \( \text{NP} \)

\( \text{RNP} \) \( \subseteq \) \( \text{NP} \)
Boolean Expression - n variables - \(2^n\) lines in truth table

\[ x \land (\neg y \lor z) \]

Satisfiable?

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(x)</td>
<td>(y)</td>
<td>(z)</td>
</tr>
<tr>
<td>(f)</td>
<td>(f)</td>
<td>(f)</td>
</tr>
<tr>
<td>(f)</td>
<td>(f)</td>
<td>(t)</td>
</tr>
<tr>
<td>(t)</td>
<td>(f)</td>
<td>(f)</td>
</tr>
<tr>
<td>(t)</td>
<td>(t)</td>
<td>(t)</td>
</tr>
<tr>
<td>(t)</td>
<td>(t)</td>
<td>(f)</td>
</tr>
<tr>
<td>(t)</td>
<td>(t)</td>
<td>(t)</td>
</tr>
</tbody>
</table>

Homework Problem in NP - Satisfiability of Boolean expressions