\[ IC \in \text{P} \quad \text{an oracle} \]

\[ \text{If SAT can be solved with ORACLE} \]

\[ \text{NP} \subseteq \text{P}^\text{NP} \quad \text{then IC can be solved in polytime} \]

\[ \text{w.r.t. SAT (oracle)} \]

\[ \text{ORACLE gives ans in 1 step.} \]

\[ \Pi_2^P = \Pi_1^P \quad \text{means we can solve } \Pi_2^P, \text{ if we solved } \Pi_1^P. \]

\[ \text{Qs. whether, } \text{NP}^\text{A} \triangleq \text{P}^\text{A} \text{, where } \text{A is ORACLE} \]

\[ \text{Nobody can prove } \text{NP} = \text{P} \text{ by diagonalisation, which is a} \]

\[ \text{ORACLE.} \]

\[ \text{Tuesday, 21st April} \]

\[ \text{Inside P} \]

\[ \text{Problem: Polynomial time may still mean too long.} \]

\[ \text{Solution: distribute, parallelize.} \]

\[ \text{Example of parallelization:} \]

\[ x_1 + x_2 + x_3 + \ldots + x_n \]

\[ \text{Let's we have 2 processors,} \]

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

\[ \text{Finally, add them together} \]

\[ x_1, x_2, x_3, x_4 \]

\[ 2 \text{ cycles} \quad n = 4 \quad t = 2 \quad p = 2 \]

\[ x, x_3, x_4, x_5, x_6, x_7, x_8 \]

\[ 3 \text{ cycles} \quad n = 8 \quad t = 3 \quad p = 4 \]

\[ x_1, x_2, \ldots, x_n \]

\[ \log_2 n \quad p = n/2 \]

\[ \text{For product, this is exactly same as} \]

\[ \text{Qs. 16 variables product: determine} \]
**Dot Product**

\[ x_1, \ldots, x_m \]
\[ y_1, \ldots, y_m \]

\[ x \cdot y = x_1 y_1 + \cdots + x_m y_m \]

\[ t = 1 + \log_2 m \]

\[ p = m \text{ for the whole product,} \]

\[ n = 2^m, \quad t = \log n, \quad p = n/2 \]

\[
\begin{pmatrix}
  a_{11} & \cdots & a_{1m} \\
  a_{21} & \cdots & a_{2m} \\
  \vdots & \ddots & \vdots \\
  a_{m1} & \cdots & a_{mm}
\end{pmatrix}
\begin{pmatrix}
  b_{11} & \cdots & b_{1m} \\
  b_{21} & \cdots & b_{2m} \\
  \vdots & \ddots & \vdots \\
  b_{m1} & \cdots & b_{mm}
\end{pmatrix}
\]

\[
\begin{pmatrix}
  c_{11} & \cdots & c_{1m} \\
  c_{21} & \cdots & c_{2m} \\
  \vdots & \ddots & \vdots \\
  c_{m1} & \cdots & c_{mm}
\end{pmatrix}
\]

\[ c_{ij} = a_{i1} b_{1j} + a_{i2} b_{2j} + \cdots + a_{im} b_{mj} \]

For each \( c_{ij} \), we need \( (1+\log_2 m) \) time in \( m \) processors.

Since we do \( n \) of them,

computing all elements of \( C \) is:

\[ t = 1 + \log_2 m \]

\[ p = m \]

\[ n = 2^m \text{ processors} \]

\[ n \frac{n}{2} \text{ elements} \]

\[
\begin{pmatrix}
  a_{11} & a_{12} \\
  a_{21} & a_{22}
\end{pmatrix}
\begin{pmatrix}
  b_{11} & b_{12} \\
  b_{21} & b_{22}
\end{pmatrix}
\]

\[
\begin{pmatrix}
  c_{11} & c_{12} \\
  c_{21} & c_{22}
\end{pmatrix}
\]

\[ n \text{ (input size)} \]

\[ n = 2^m \]

\[ m = n^{1/2} \]

\[ p = n^{3/2} \]
Parallelizable

A problem is parallelizable if we can solve it in polylog time on polynomial # f processors:

\[ t \leq P(\log n) \]

**NC (Nick's class)**

If we have \( p \) processors \( t \) cycles (for sequential calculation)

\[ t \leq p \]

\[ p \leq P_1(n) \]

\[ t \leq P_2(\log n) \]

Multiplication is also bounded by polynomial.

**NC \subseteq P**

So, NC is subset of P.

**\( P^2 = NC \)**

\( P \text{-complete} = \) problems hard in \( P \) if they can be parallelized, then everything else can be.

Ex. of \( P \)-complete problem: linear programming (LP)

Finding solutions to systems of linear inequalities

<table>
<thead>
<tr>
<th>Food</th>
<th>( X_1 )</th>
<th>( X_2 )</th>
<th>( X_3 )</th>
<th>( X_4 )</th>
<th>( X_5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calories</td>
<td>2000</td>
<td>4000</td>
<td>100</td>
<td>1000</td>
<td>150</td>
</tr>
<tr>
<td>Protein</td>
<td>2.000 ( X_1 ) + 4.000 ( X_2 ) + 100 ( X_3 ) + 1000 ( X_4 ) + 150 ( X_5 ) ( \geq ) 2500</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vitamin</td>
<td>0.1 ( X_1 ) + 0.2 ( X_2 ) + 0.3 ( X_3 ) + 0.4 ( X_4 ) + 0.5 ( X_5 ) ( \geq ) 0.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cost</td>
<td>1 ( X_1 ) + 2 ( X_2 ) + 3 ( X_3 ) + 2 ( X_4 ) + 7 ( X_5 ) ( \leq ) 9.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(To meet this conclusion, we can find \( x_1, \ldots, x_5 \))

How do you solve this problem? We try 2^n possible variations.

Inequality.
Simplex method: \(2^n\) time is required to solve LP.

Khachiyan solved LP \(\leq P\).


**Theorem:** LP is \(\text{NP-complete} \).

Ideal world, when we take into accont-time \(t \sim \log n\), parallel.

In practice??

\[ T_{\text{seq}} \Rightarrow T_{\text{par}} \]

\[ R = C \cdot T_{\text{par}} \quad V = \frac{4}{3} \pi R^3 \]

\[ = \frac{4}{3} \pi C^2 \cdot T_{\text{par}}^3 \text{ constant} = C \]

\[ \text{we are here} \quad N_{\text{processor}} \leq \frac{V}{4U} = \frac{4}{3} \pi C^3 \cdot T_{\text{par}}^3 \]

\[ T_{\text{seq}} \leq N_{\text{processor}} \cdot T_{\text{par}} \leq C \cdot T_{\text{par}}^3 \cdot T_{\text{par}} \]

\[ T_{\text{par}} \geq C \cdot T_{\text{seq}} \]

\[ T_{\text{par}} \geq C \cdot T_{\text{seq}} \]

\[ \frac{4}{3} \geq C \cdot T_{\text{par}} \]

\[ \frac{4}{3} \geq C \cdot T_{\text{par}} \]

\( T_{\text{par}} \geq C \cdot T_{\text{seq}} \)

\( C \cdot T_{\text{par}} \) (it's not good as \( \log n \))

we have limitations, caused by:

1. \( n \leq C \).

2. \[ V = \frac{4}{3} \pi R^3 \] Euclidean geometry.

   real space-time is curved.

  4th postulate.

  There's one line \( l \) to \( l \) through point \( a \).