Def: $f: \{0,1\}^* \rightarrow \{0,1\}^*$ is implicitly logspace computable if any bit of $f(x)$ can be computed in time $O(\log |x|)$ space.

Formally, $\exists c$ s.t. $|f(x)| \leq |x|^c$ $\forall x$.

$\exists \bar{c} = \bar{\langle x, i \rangle} \mid f(x)_i = 1^2$ and $\exists \bar{c}' = \bar{\langle x, i \rangle} \mid i \leq |f(x)|^3$

are in $\mathcal{L}$.

$B \leq_L C$ (B is log-space reducible to C)
if $\exists f$ that is implicitly logspace computable st $f(x) \in C$ if and only if $x \in B$.

$C$ is INL complete if $C \in \text{INL}$ and $\forall B \in \text{INL}, B \leq_L C$.

Lemma: (1) If $B \leq_L C$ and $C \leq_L D$, $B \leq_L D$.
(2) If $B \leq_L C$ and $C \in \text{ELL}$, $B \in \text{ELL}$.

Pf: Let $f$ reduce $B$ to $C$ and $g$ reduce $C$ to $D$.
We have to show $g(f(x))$ can be computed in logspace.

At any stage $M_g$ only needs a particular bit of $f(x)$.

Modify $M_g$ as follows.

Other than oracle everything is logspace.
The oracle is basically $M_f$: 

\[ \text{Pos. } \xrightarrow{\text{input}} \text{Oracle } \xrightarrow{f(x)} M_x \xrightarrow{f(x)} \]

\[ \text{Total storage } = O(S_f(1x) + S_g(1f(x)) + \log \log f(x)) \]

\[ = O(\log n) \]

Thus: \[ \text{PATH} \notin S \] if $G$ is directed graph and there is a path from $s$ to $t$.

\[ \mathcal{P} \text{ PATH is \textbf{NL-complete}.} \]

Proof: \[ \text{PATH} \notin \text{NL}. \text{ Just non-deterministically guess the path. Only needs to store current vertex.} \]

\[ L \leq \text{NL} \] \[ M \text{ decides } L \text{ in } O(\log n) \text{ space } \]

\[ \text{Convert } \# \text{ to } x \text{ to } G_{M,x} \text{ is logspace. } \]

\[ M \text{ is easy to convert } x \text{ to } G^\text{start} \text{ and } G^\text{accept} \text{ in logspace (since } M \text{ is } O(\log n) \text{ space machine.)} \]

\[ G_{M,x} \text{ adj. matrix } = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} \text{ depending on } (c, c') \]

Given $c, c'$, this bit can be determined in logspace.
The certificate viewpoint

Can we define INP in terms of certificates?

Naïve defn: \( L \subseteq INP \) if \( \exists \) poly \( p \) & logspace machine \( M \)

\[ \forall x \in L \iff \exists y, |y| \leq p(|x|) \text{ s.t. } M(<x,y>) \text{ accepts.} \]

This doesn't work!

Such a machine/setting can solve 3SAT!

Given 3CNF \( \Phi \) and possible assignment \( z \in \{0,1\}^* \),
we can check if \( \Phi(z) = 1 \) in logspace:

\[ z_i \overrightarrow{z_j} \overrightarrow{z_k} \rightarrow M \quad \text{Work} \]

For each \( z_i \): \( M \) finds values of \( z_i, z_j, z_k \). It only
needs \( O(\log n) \) space to store index \( i, j, k \).

Also, (looking carefully at Cook-Levin reduction)
\[ \forall L \in \text{NP}, \quad L \subseteq \text{SAT} \]

Thus, \( L \in \text{INP} \) if \( \exists \) poly \( p \) & logspace machine \( M \)

\[ \text{st.} \quad - - - - - - - \]

This characterizes INP. You don't need \( M \) to be poly time.
(for Cook-Levin reduction)

\[
\Phi = \text{UNIQUE} \land \text{START} \land \text{ACCEPT} \land \text{MOVE}
\]

\[\text{Independent of input } x!\]

So the \(i\)th bit is just for some fixed function of \(i\)

\[
\text{SA START } = x_{0000} \land x_{0001} \land \ldots
\]

dependence on input, can be determined in logspace.

The right definition

Define a tape to be read-once if head never writes on tape and only moves never moves left.

\(L \in \text{NL} \iff \exists \text{poly } p \exists \text{ logspace machine } M \text{ st.}
\]

\[x \in L \iff \exists y, |y| \leq p(|x|) \text{ st.}
\]

\[M(x \downarrow y) \text{ accepts where } x \text{ is on input tape any and } y \text{ is on read-once tape.}
\]

Thm: This characterizes NL.

Pf: Suppose \(L \in \text{NSPACE}(|x|) \Rightarrow \text{NL}\)

Consider NTFM M deciding \(L\).

Define logspace machine M that reads off non-deterministic choices from a separate tape. This tape has certificate \(y\), and head on this tape only moves right.