- Review of Exam 2
- L, NL, NL-Completeness (Sipser 8.4-8.5)

Some material from slides by M. Sipser
Announcements

- Exam 2 is graded.
  - Mean:
  - Median:
  - Std Dev:

- There will only be one more homework
  - Homework 5: Assigned 4/8/21, Due 4/23/31
  - Homework 6 is cancelled
Each student needs to give a 30 – 35 minute presentation
- Dates: April 26 and April 29
- If you cannot attend class live, contact me to discuss alternatives

Possible topics
- Hypercomputation – Infinite computation
  - Includes computation near a black hole
- Hypercomputation – Turing’s oracle machines
- Neural Turing Machines
- DNA Computing
- Analog Computing

Google doc in Submitty Course Materials with suggested references for each topic (will be posted after class today)
- Email me your top 3 choices (in order) and I will do my best to accommodate
- You can also select your own topic – just clear it with me in advance
Are there problems that can be solved in sublinear space, \( f(n) < n \) ?

To study sublinear space algorithms, we need a different model.
- Two-tape TM with read-only input tape and read/write work tape.

*read-only input tape does not count towards space used*

*count cells used here*

*read/write work tape*
Log Space Complexity

- We focus on algorithms with space complexity in $O(\log n)$

- $L$ is the class of languages that are decidable in logarithmic space on a deterministic TM:
  \[ L = SPACE(\log n) \]

- $N$ is the class of languages that are decidable in logarithmic space on a nondeterministic TM:
  \[ NL = NSPACE(\log n) \]
Example in Class L

- \( A = \{0^k1^k \mid k \geq 0 \} \)

- D: on input \( 0^k1^k \):
  - Count number of 0’s and 1’s using two binary counters on work tape
  - Compare counters, if equal, accept. If not, reject.
  - (Implicit: If a 0 appears after a 1, reject)
A = language of properly nested parenthesis. Show that A is in L
The PATH Problem - Revisited

- \( \text{PATH} = \{ \langle G, s, t \rangle | G \text{ is a directed graph that has a directed path from } s \text{ to } t \} \)

- We showed that PATH is in P

- M: On input \( \langle G, s, t \rangle \) where \( G \) is a directed graph with nodes \( s \) and \( t \)
  - 1. Mark node \( s \).
  - 2. Repeat the following until no additional nodes are marked.
    - 3. Scan all edges of \( G \). If an edge \( (a, b) \) is found going from a marked node \( a \) to an unmarked node \( b \), mark node \( b \).
    - 4. If \( t \) is marked, accept. Otherwise, reject.
\[ \text{PATH} = \{ (G, s, t) \mid G \text{ is a directed graph that has a directed path from } s \text{ to } t \} \]

Show \( \text{PATH} \) is in NL.
Relationship to Other Complexity Classes

- **Theorem:** $NL \subseteq P$

- **Theorem:** $NL \subseteq \text{SPACE}(\log^2 n)$
NL-Completeness

- PATH problem is in NL. We do not think it is in L.
  - We don’t know whether L = NL.

- We define a language $B$ to be **NL-complete** if
  1. $B$ is in NL
  2. All other languages in NL are reducible to $B$ in log space.
A **log space transducer** is a TM with 3 tapes:

1. A read-only input tape
2. A write-only output tape
3. A read/write work tape of size $O(\log n)$

A log space transducer $T$ computes a function $f : \Sigma^* \rightarrow \Sigma^*$ if $T$ on input $w$ halts with $f(w)$ on its output tape, for all $w$.

Call $f$ a **log space computable function**.

Language $A$ is **log space reducible** to language $B$ if $A$ is mapping reducible to $B$ using a log space computable function $f$.

Written: $A \leq_L B$
Theorem: If $A \leq_L B$ and $B \in L$ then $A \in L$. 
A language $B$ is **NL-complete** if
1. $B$ is in $NL$
2. For all $A \in NL$, $A \leq_L B$.

Theorem: If any NL-complete language is in $L$, then $L = NL$. 
Theorem: PATH is NL-Complete