I/O-comparison trees

[Arge, Knudsen, Larsen, 93]
Goal:
A general reduction theorem:

Lower bound on comparisons to solve a problem

Lower bound on I/Os to solve the problem

Method:
Extend the notion of comparison trees.
Standard Comparison Trees

- Binary trees.
- Internal node labelled with pairs of elements, represents comparisons.
- Edges labelled with one of the possible outcomes of the comparison above.
- Leaves labelled with one possible answer to problem ("Yes/No" for decision problems, a permutation for construction problems, an element for search problems)

Tree solves a problem

∀ leaves l: ∀ input x
ending in l: label of l is correct for x.
I/O Comparison Trees

- Add unary I/O-nodes to comparison trees.
- I/O node labelled with position in memory of all elements before and after I/O.
- Root and leaves: I/O-nodes.
- Comparison nodes may only compare nodes in RAM (given by label of lowest ancestor which is an I/O-node).
Compression

Compress comparison-only subtrees:

$T$: minimal height comparison tree to sort contents of RAM at $v$. 
Reduction

Compress entire tree by compressing all comparison-only subtrees in top-down order:

\[ T_1 \rightarrow T_2 \]

By induction on number of I/O-nodes on path: an input \( x \) will pass exactly the same I/O-nodes (same number of nodes having the same labels) in \( T_1 \) and \( T_2 \).

Corollary: \( x \) ends up in leaf with same label in \( T_1 \) and \( T_2 \).

Finally, remove all I/O-nodes from \( T_2 \):

\[ T_2 \rightarrow T_3 \]

Now \( T_3 \) is standard comparison tree solving same problem.
**Height of** $T$

**Theorem:** Comparison complexity of sorting $n$ elements is $\Theta(n \log n)$.

**Theorem:** Comparison complexity of merging two sorted lists of lengths $n$ and $m$ is $\Theta(m(\log(n/m) + 1))$, assuming $n \geq m$.

At most $N/B$ untouched blocks.
Reduction Analysis

∀ inputs x:

\[ |\text{path in } T_3| = |\text{sti i } T_2| - |I/Os in T_2| \]
\[ \leq [I/Os in T_2] \cdot (B \log(M/B) - 1 - 1) + (N/B)B \log B \]
\[ \leq [I/Os in T_2] \cdot B \log(M/B) + (N/B)B \log B \]
\[ \leq [I/Os in T_1] \cdot B \log(M/B) + N \log B \]

∃ comparison lower bound \( L \Rightarrow L \leq |\text{path in } T_3| \)

\[ \frac{L - N \log B}{B \log(M/B)} \leq I/Os \text{ in } T_1 \]
Examples

\[
\frac{L - N \log B}{B \log(M/B)} \leq \text{I/Os in } T_1
\]

<table>
<thead>
<tr>
<th>Problem</th>
<th>(L)</th>
<th>I/O Lower Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sorting</td>
<td>(N \log N)</td>
<td>((N/B) \log_{M/B}(N/B))</td>
</tr>
<tr>
<td>Set equality</td>
<td>(N \log N)</td>
<td>do.</td>
</tr>
<tr>
<td>Set inclusion</td>
<td>(N \log N)</td>
<td>do.</td>
</tr>
<tr>
<td>Set disjointness</td>
<td>(N \log N)</td>
<td>do.</td>
</tr>
</tbody>
</table>

Multiset sorting, duplicate removal, mode finding: see paper.