<table>
<thead>
<tr>
<th>Operation</th>
<th>Rep A (List)</th>
<th>Rep B (Ordered List)</th>
<th>Rep C</th>
<th>Rep D</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>add(x) x ∈ S</code></td>
<td>~n</td>
<td>~n/2</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>contains(x) x ∈ S</code></td>
<td>~n</td>
<td>~n/2</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>remove(x) x ∈ S</code></td>
<td>~n/2</td>
<td>~n/2</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>intersect(S)</code> S ∩ T = ∅</td>
<td>~n · m</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>l T l = m</code></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Rep A: Unordered List

\text{add}(x) \quad \text{traverse the entire list, stopping if } x \text{ is found}
\quad \text{if } x \text{ was not found, add it at the end}

\text{contains}(x) \quad \text{check all items for equality to } x

\text{remove}(x) \quad \text{find list item for } x
\quad \text{if found it, fix up pointers}

\text{intersect}(T) \quad \text{take the smaller list}
\quad \text{for each element } e \text{ in larger list, 2}
\quad \text{if } e \text{ in larger list, put } e \text{ in the result
for each element $e$ in $S$ \[ \text{if } T \text{ contains } e \begin{cases} \text{put } e \text{ into result set} \\ \text{time}=m \end{cases} \] $n \times m$
Sorted List: no duplicates, ordered increasing

- **add(x)**: traverse the list until either finding x, a larger item, or the end. If x was not found, insert x before final position.
  \[2, 4, 7, 9, 16, 23\] ~ n/2

- **contains(x)**: look through the list until either finding x or a larger item or the end.
  \(~ n/2\)

- **remove(x)**: look through the list until finding x (take it out) or a larger item (do nothing) or tail (do nothing).
  \(~ n/2\)
Intersection for sorted lists:

**Approach 1:** Use same alg:

\[ n \times \frac{m}{2} \]

for (E e : S)
if T.contains(e)
R.add(e)

Rev: for (E e : S)
if T.contains(e)
time l \leftarrow R.addAtEnd(e)

\[ \sim M+N \]

\[ \{5, 9, 16\} \]