CSCI 241
Scott Wehrwein
Interface vs Implementation
Abstract Data Types
Goals

• Understand the distinction between interface and implementation in the context of operations on data structures.

• Know the meaning of abstract data type.
Last time: Big-Deal CS Concept #1: Asymptotic Runtime Analysis
Today: Big-Deal CS Concept #2: Interface vs Implementation and Abstract Data Types

**What** the operations do

An abstract data type specifies only interface, not implementation

**How** they are accomplished
Interface vs Implementation: Example

(interface) **Drawer**

- **FilingDrawer** (Implementation 1)
- **PilingDrawer** (Implementation 2)
Interface vs Implementation: Example

**Interface**

- Contains(item) - returns true iff item is in the drawer
- Add(item) - adds item to the drawer
- Remove(item) - removes item from the drawer if it exists

**Implementation**

FilingDrawer implements Drawer:

Contains(item):

*binary search drawer for item*
Comparing Implementations

class FilingDrawer:
  • Contains(item):
    
    binary search drawer for item \(O(\log(N))\)

class PilingDrawer:
  • Contains(item):
    
    linear search drawer for item \(O(N)\)
Comparing Implementations

class FilingDrawer: \(O(N)\)

• Add(item):
  insert item in sorted position in drawer

class PilingDrawer: \(O(1)\)

• Add(item):
  insert item into the drawer wherever it's easiest
Which is better?

<table>
<thead>
<tr>
<th></th>
<th>PilingDrawer</th>
<th>FilingDrawer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>O(1)</td>
<td>O(N)</td>
</tr>
<tr>
<td>Find</td>
<td>O(N)</td>
<td>O(log N)</td>
</tr>
</tbody>
</table>

The best choice of implementation for an abstract data type usually depends on how you'll be using it.
## Abstract Data Types: Examples

<table>
<thead>
<tr>
<th>ADT</th>
<th>Notable Implementations</th>
<th>When is it covered?</th>
</tr>
</thead>
<tbody>
<tr>
<td>List, Queue, Stack</td>
<td>Array, Nodes</td>
<td>(CSCI 145)</td>
</tr>
<tr>
<td></td>
<td>Array, Nodes</td>
<td>(Weeks 4,5,7)</td>
</tr>
<tr>
<td>Set</td>
<td>Array, Nodes, Tree, Hash table</td>
<td>(Weeks 4-6; A2)</td>
</tr>
<tr>
<td>Tree</td>
<td>Array, Nodes</td>
<td>(Week 6; A3)</td>
</tr>
<tr>
<td>Priority Queue</td>
<td>Heap</td>
<td>(Week 7; A3)</td>
</tr>
<tr>
<td>Map</td>
<td>Tree, Hash table</td>
<td>(Weeks 8-9; A4)</td>
</tr>
<tr>
<td>Graph</td>
<td>Array, Nodes</td>
<td></td>
</tr>
</tbody>
</table>
Philosophical Question:

Is an array an ADT?