Taxonomy of ADTs (Secs 2.1-2.6)
Disjoint Set of Elements

Collection of Elements

Partition

We'll focus on these for much of this course (and much of the text).

Binary relationships (methods involve looking at paths)

Graphs
ex) print queue in order submitted → Manually Positioned

Collection of Elements

Algorithmically Positioned

ex) queue ordered by the priority for each job
Manually Positioned Collections

0 1 2 \( \ldots \) \( n-1 \) positions
Positional Collection (Chapter 9, pages 107-119)

position 0 1 ... n-1

array-based implementation

+ constant time access to position p
- must shift elements
- to add or remove near middle (linear time)

list-based implementation

- must traverse list to reach position p
- linear time to get to middle
+ constant time to add or remove once element located
array-based

- some space (and time) overhead to maintain a tracker

+ space usage can be as small as roughly $n$ references when $n$ elements in collection

- must select a size when allocating (you can resize)

list-based

+ negligible overhead to track an element

- even in a singly-linked list use 3 refs per node (element, next, type)

Roughly 3n references to hold same n elements

+ naturally can dynamically resize
Positional Collection Data Structures

- Array position p stored in a[p]

- Dynamic Array
  (This is what a Java arraylist is!)

Initial size

Insertion at ystrip[4] triggers resizing

Changes when doubling array size
Circular Array

Queue with at most 4 elements

Offset of 1, let variable $\text{start} = 1$ to mark index for position 0

degqueue (to remove a)

Translation between index + position computed by:

\[
\text{index} = \text{position} + \text{start} \mod \text{array size}
\]

\[
\text{position} = \text{index} - \text{start} \mod \text{array size}
\]
index

offset into underlying array for position 0

start = 8

size = 7

Queue: f, g, h, i, j, k, l

Each dequeue implicitly changes positions of remaining elements by incrementing start
Array
Dynamic Array
Circular Array
Dynamic Circular Array

Tracked Array

application program must maintain a reference to the tracker returned by addTracked

Singly Linked List
Doubly Linked List
Table 9.1 in *A Practical Guide to Data Structures and Algorithms in Java*

For time $O(1) \sim O(\min(p+n-p)) = O(n)$

<table>
<thead>
<tr>
<th>Operation</th>
<th>Array</th>
<th>Circular Array</th>
<th>DynamicArray</th>
<th>Dynamic-Circular Array</th>
<th>TrackedArray</th>
<th>SinglyLinkedList</th>
<th>DoublyLinkedList</th>
</tr>
</thead>
<tbody>
<tr>
<td>access by position near middle</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>adding to middle portion (once located)</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>removing from middle portion (once located)</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>access to front</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>adding to front</td>
<td>●</td>
<td>○</td>
<td>●</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
</tr>
<tr>
<td>removing at front</td>
<td>●</td>
<td>○</td>
<td>●</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
</tr>
<tr>
<td>access to back</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>adding to back</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>removing from back</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>determining the position of a locator</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>support of a tracker</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>space usage</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
</tr>
<tr>
<td>ease in adjusting the capacity</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>○</td>
<td>●</td>
</tr>
<tr>
<td>automatic resizing</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>