Dynamic Array Queue and Deque
int isEmpty();

void addBack(TYPE val);   // Add value at end of queue.

TYPE front();             // Get value at front of queue.

void removeFront();       // Remove value at front.
Queue Applications

• Also good for ‘remembering’, just in a different order. We’ll revisit this when we study graphs and search!
• Discrete event simulations
• Operating systems
Queue with Dynamic Array

int isEmpty();
void addBack(TYPE val); // Add value at end of queue.
TYPE front(); // Get value at front of queue.
void removeFront(); // Remove value at front.
Deque (Double Ended Queue)

void addFront(TYPE val);
void removeFront();
TYPE front();
void addBack(TYPE val);
void removeBack();
TYPE back();
Dynamic Array Deque

Adding to Front

Removing from Front
One solution: Rather than always use index zero as our starting point, allow the starting index to “float”

- Maintain two integer values:
  - Starting or beginning index \( (\text{beg}) \)
  - Count of elements in the collection \( (\text{size}) \)

- Still need to reallocate when size equal to capacity
Dynamic Array Deque

- First filled element no longer always at index 0
- Filled elements may wrap around back to the front end of array
- Called ArrDeque or IndexedDeque
Deque Application

- Finite Length Undo
Your Turn

• Read Worksheet 20 Introduction