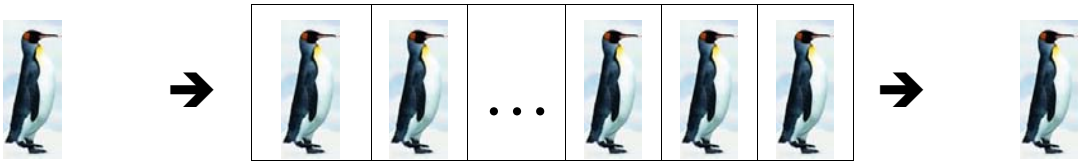
Queues

- Objectives:
  - Real life examples
    - Waiting in line
    - Waiting on hold for tech support
  - Applications related to Computer Science
    - Threads
    - Job scheduling

- **Definition:**
  - It is an FIFO ADT
  - A new element is added or inserted to the end of the list
  - An element is deleted or removed only from the beginning of the list.



Enqueue Dequeue

- **Operations:**
  - Isempty
  - Enqueue (Insert)
  - Dequeue (Delete)
  - Etc.

- **Implementations**
  - Arrays
  - Linked lists

- Time Complexity:

Big-O Comparison of Queue Operations			
Operation	Array Implementation	Singly Linked Implementation	Linked List with Head and Tail Implementation
Class constructor	O(1)	O(1)	O(1)
MakeEmpty	O(1)	In Java, O(1) Others: O(N)	In Java, O(1) Others: O(N)
IsFull	O(1)	O(1)	O(1)
IsEmpty	O(1)	O(1)	O(1)
Enqueue	O(1)	O(1)	O(1)
Dequeue	O(1)	O(N)	O(1)
Destructor	O(1)	O(1)	O(1)

- A double-ended queue, or deque,
  - It supports insertion and deletion from the front and back
  - Modeling entities (people, cars, etc.) in real-world waiting queue
    - Entering at the queue
    - Waiting entities may decide to leave the line
  - The deque supports the following operations:
    - Insert First: Inserts e at the beginning of deque
    - Insert Last: Inserts e at end of deque
    - Remove First: Removes the first element
    - Remove Last: Removes the last element

- priority queue:
  - It is a queue where each element is associated with a key (priority) value at the time it is inserted.
    - It has two main operations:
      insert element(k,e): Insert an element e with key k into the queue.
    - removeMin()/removeMax(): Return and remove from the queue an element with the min/max key.