Stacks

September 26, 2003

Stack functions

- **push(item)**: adds an element to the top of the stack
  - $O(1)$
- **pop()**: removes an element from the top of the stack
  - $O(1)$
- **peek()**: retrieves the data from the topmost element on the stack, but leaves it on the stack

Stack implementation

- Linked list: preferred
  - push: `addToHead(...)`
  - pop: `removeFromHead(...)`
- Vector
  - first element = oldest (tail)
  - last element = newest (head)
  - pop: `remove([index of last item])`
  - push: `add(...)`
Java's Stack class

- in java.util
- built-in push, pop, and peek functions
- subclass of Vector
- Not a “pure” stack
  - we can technically access elements other than the topmost one with this implementation

Queue

- “First-in, first-out” (FIFO) structure
- Items are added to the tail and removed from the head
- Examples: lines at banks, stores; call banks at service centers; page requests at a web server; traffic at an intersection; etc.
- Widely used in simulation

Properties of queues

- Arrival rate
  - how often, and in what patterns, people/objects/etc enter the queue

- Service time
  - how quickly is each person/object/etc in the queue served

- Waiting time
  - how long a person/object/etc waits to be served

- Capacity
  - how long is the line allowed to get?

Queue functions

- enqueue(item): adds an item to the end of the queue
  - O(1)

- dequeue(): removes an item from the beginning of the queue
  - O(1)

- peek(): retrieves the data from the first item in the queue without removing it
Queue implementation

- Single linked list: preferred
  - enqueue: addToTail(...)
  - dequeue: removeFromHead()
- Array
  - “circular array”
  - first and last items in queue may be stored somewhere other than the first and last slots in the array
  - need “pointers” to first and last item

Priority queues

- Queues with different dequeue() methods for various items in the queue
- Examples: first-class check-in lines at the airport, I-Pass lanes at tollbooths, “15 items or less” lines at the supermarket, “FastPass” lines at amusement parks
- Items in the queue are of different service classes
  - higher-ranked service classes have “priority”, meaning that they are processed before lower-ranked service classes

Implementing priority queues

- Linked list
- Option 1: items are stored in the order in which they enter
  - dequeue(): search for the first item of highest priority, and remove it
  - O(n)
- Option 2: items are sorted by priority as they arrive, and removed from front to back
  - O(n)
- Several other (more complex) options...

Fun with queues

- There is no built-in queue class in Java
- There are many different types of queues:
  - multi-server queues (e.g., tellers at a bank with a single line of customers)
  - preemptive queues: items may reenter the queue in the middle of being served
  - many other ways to order and serve items in a queue: LIFO, shortest-service-time-first, highest-payoff, etc.
- An entire field of math/engineering/CS is devoted to the theory of queues