DS.L.1

Lists, Stacks, and Queues (Review)

Chapter 3 Overview

- **Abstract Data Types**
- **Linked Lists, Headers, Circular Links**
- **Cursor (Array) Implementation**
- **Stacks (Linked and Array)**
- **Stack Applications**
- **Queues**
- **Queue Application**

Read and review! Much of this should be familiar.

DS.L.2

A list is an ordered sequence of elements A1, A2, . . . , AN.

Array Implementation:

0	size = N
1	A1
2	A2
.	.
.	.
N	AN
max	

Size = N
Max = M

or

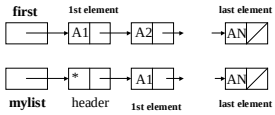
0	A1
1	A2
.	.
.	.
N-1	AN
max	

What is the complexity for common operations?

- Advance (to next)
- MakeEmpty
- IsEmpty
- Insert/Delete
- Find
- Traverse

DS.L.3

Linked Lists



Variations:

circular
doubly linked
cursor implementation

DS.L.4

Cursor Implementation of Linked Lists

C and C++ have explicit dynamic allocation and freeing of storage.

malloc	free
new	free

Lisp and Java have dynamic allocation and use garbage collection for automatic freeing.

What do we do about FORTRAN?

- use an array with fields for data and links
- make the links be integer subscripts

DS.L.5

Stacks

- operations: push, pop, top
- can be array or linked
- complexity O(1) for most everything

Queues

- operations are enqueue, dequeue, front, rear
- usually linked with front and rear pointers
- complexity O(1) for most everything

DS.L.5

Stack Applications

- function calls
- nonrecursive recursion
- postfix expression evaluation
- conversion from infix to postfix

Queue Applications

- process queue in operating system
- event queue in simulation
- also used in infix to postfix