List - linear data structure

Ming-Hwa Wang, Ph.D.
COEN 279/AMTH 377 Design and Analysis of Algorithms
Department of Computer Engineering
Santa Clara University

Operation

- basic operations:
  - first(*L*)
  - end(*L*)
  - next(*p*, *L*)
  - locate(*x*, *L*)
  - retrieve(*p*, *L*)
  - insert(*x*, *p*, *L*)
  - delete(*p*, *L*)
  - makenull(*L*)
  - printlist(*L*)
- derived operations
  - purge(*L*)

Implementation

- array
  - size has to be predefined, random access
  - one dimensional array
  - two dimensional array
    - row-major ordering
    - column-major ordering
  - multiple dimensional array
- pointer or linked list
  - size can grow or shrink, sequential access
  - w/ or w/o head/sentinel node
  - singly-linked lists
  - doubly-linked lists
  - circular linked lists
  - multilist
  - applications
    sparse matrices
- cursor implementation of linked list for languages have no pointer type
  - multiple array representation: represent a collection of objects that have the same field by using an array for each field, then a pointer is simply a common index into the (key, next, prev) arrays
  - Single-array representation: a pointer is simply the address of the first memory location of the object, and other memory locations within the object can be indexed by adding an offset to the pointer (it permits heterogeneous objects of different lengths)
- comparisons
  - static vs. dynamic allocation
  - sequential vs. random access
  - operation run-time efficiency
  - memory efficiency

Stack - LIFO

- operations
  - makenull(*S*)
  - top(*S*)
  - pop(*S*)
  - push(*x*, *S*)
  - empty(*S*)
- applications
  - activation record
  - multiple stacks in memory management
  - prefix, infix, postfix and conversions
    prefix and postfix don't need parenthesis

Queue - FIFO

- operations
  - makenull(*Q*)
  - front(*Q*)
  - tail(*Q*)
  - enqueue(*Q*)
  - dequeue(*Q*)
  - empty(*Q*)
- circular queue
- priority queue
- applications

Deque

combined stack and queue

Mapping - associated array

- content addressable
- operations
  - makenull(*M*)
  - assign(*M*, *d*, *r*)

- compute(M, d. r)
- applications