# CSE 674 Advanced Data Structures

## Lists

Andrew C. Lee

EECS, Syracuse

# Contents

- Lists
- Sequential and Linked Allocation
- Pointers
- Singly linked lists and doubly linked lists
- Stacks and Queues

# Lists

- Representing data via records
- Each records may have different fields (data may be of different types)
- Representing the aggregration of data by a *list* or a *set*

**Question**: What's the difference(s) between a *list* and a *set* ?

# Sequential and linked allocation

- Sequential allocation via arrays
- Arrays can have more than 1 dimensions
- Linked Allocation: data fields contains addresses of other records
- Examples: singly linked lists, doubly linked lists

**Question**: What are the key differences between *sequential* allocation and *linked* allocation ?

# Pointers

**Discussions**: What does the program `pointers.cpp` do ?

# Singly linked lists and Doubly linked lists

1. Singly linked lists

   ```
   myList -->[1] --> [2] -->[3] -->NULL
   ```

2. Doubly linked lists

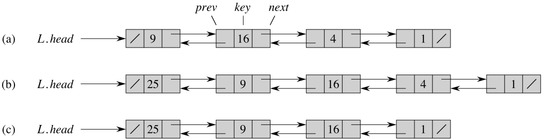

Figure : Sketches of a Doubly linked lists from Cormen's text

**Discussions**: Describe any major difference(s) between the above two data structures.

# Stacks and Queues

- Stacks: Last In First Out (LIFO)
- Main operations of a Stack: push, pop
- Queues: First In First Out (FIFO)
- Main operations of a Queue: enqueue, dequeue

Both of them can be implementation via sequential or linked allocation strategies

# Suggested Readings

1. Run code examples: `pointers.cpp`
2. Read Brass, Chapter 1.
3. Read CLRS, Chapter 10, Section 1 to 2.