CS2123 Data Structures
LinkedSet

Dr. Weining Zhang

Department of Computer Science
University of Texas at San Antonio

August 23, 2009

Outline

1. LinkedSet
   - SetADT
   - Linked Implementation of SetADT
   - Complexity of LinkedSet
   - Other Implementation Options

Tasks of the Week

- Case Study: Implement the SetADT collection using a linked list
- Implement an ordered linked list
- Use link structures to solve simple problems
- Compare implementations using variations of linked structures

SetADT

```
public interface SetADT<T> {
    public void add(T element);
    public void addAll(SetADT<T> set);
    public T removeRandom();
    public T remove(T element);
    public SetADT<T> union(SetADT<T> set);
    public boolean contains(T target);
    public boolean equals(SetADT<T> set);
    public boolean isEmpty();
    public int size();
    public Iterator<T> iterator();
    public String toString();
}
```

Linked Set

Implement Set ADT

- Store each element in a LinearNode object and link LinearNodes into a list
- Each LinearNode has a reference to Element and to next LinearNode
- The LinkedSet has a reference to the first LinearNode and has a count
- Always add new element at the beginning of the linked list
- Provide an Iterator which returns the next element in the linked list.
  - The Iterator has references to the linked list, the current and the next LinearNodes

Complexity of Methods

Example

In LinkedSet class,
- size(): \( O(1) \)
- add(): test size \( O(1) \), place new element at the beginning of linked list \( O(1) \)
- contains(): linear search for the element \( O(n) \)
- toString(): print each element \( O(n) \)

Using a Dummy Header Node

- Make the first LinearNode a dummy header, which does not refer to any element.
- This can simplify add and remove methods because the list always has at least one node (not element).
DoublyLinkedSet

- Store each element in a DoubleNode, which has a reference to the element, and two references to DoubleNodes, one for the previous node and the other for the next node.

LinkedOrderedSet

- Element type must implement Comparable interface
- Store elements in ascending order
- To add a new element, we must consider following cases:
  - the linked list is empty
  - the list is not empty, but the element is smaller than the first element
  - the list is not empty, the element is the largest
  - the list is not empty and the element is not the smallest nor the largest