CS1083 Week 10: More Linked Lists

David Bremner

2018-03-14

Outline

Inserting in Sorted Order

Doubly linked lists

Self sorting lists

SortedList

```
String[] names={"bob", "rudolph", "dasher"};

SortedList<String> a=new SortedList<>();

for (int i=0; i < names.length; i++){
    a.insert(names[i]);
}

a.print();
```

```
bob
dasher
rudolph
```

Design parameters

- Singly linked list
- using ComparableNode
- support `insert` and `print`

Case Analysis

- Draw diagrams for
  - inserting first
    - empty list
    - non-empty list
  - inserting in middle
  - inserting at end
- Did we miss any cases?

Insert method

- can we collapse any of the cases?
- general case
  - loop to find position
  - reference updates

Inserting in Sorted Order

Doubly linked lists

Why doubly linked lists?

- ability to traverse forwards and backwards
- simplifies deletion and insertion in middle of list

toString for list nodes

DoubleNode

```
public String toString() {
    return "[" + data.toString() + ";" +
        formatRef(prev) + ";" + formatRef(next) + "]";
}
```

- why formatRef? Try without.

Changes to variables

DoubleNode

- need a prev reference for node class
- need a last pointer for list class

insertFirst

DoubleList

- what cases?
- reference updates

insertLast

DoubleList

- what cases?
- reference updates

printReverse

DoubleList

- what to change?

```
public void print(){
    for (DoubleNode<T> current=first; current!=null;
         current=current.getNext()){
        current.print();
    }

}
```

removeFirst

DoubleList

- what error conditions?
- what cases?
- what references need to be updated?
- return value?

removeLast

DoubleList

- what error conditions?
- what cases?
- do we need a loop?
- what references need to be updated?
- return value?

remove by reference

DoubleList

- what error conditions?
- what cases?
- what references need to be updated?

```
public void remove(DoubleNode<T> node) {
⋮
}
```

remove by key

DoubleList

- how can we search the list (without Comparable)
- use previous method to do update

```
public void remove(T key) {
⋮
}
```