ECE 2574: Data Structures and Algorithms - Linked-Based Implementations

C. L. Wyatt

Fall 2017

Today we will look at an alternative to using arrays (statically or dynamically allocated) as a data structure.

- Warmup
- Nodes
- Linking nodes into chains
- Operations
- Implementing the ADT Bag with a Linked-List

Warmup #1

Suppose you had a list of integers represented as an array.

```
int a[5]; = {12,42,56,90};
a[0] = 12;
a[1] = 42;
a[2] = 56;
a[3] = 90;
```

and you want to insert the value 15 in the list so that the values remained in order. Would you:

- replace 12 by 15
- replace 42 by 15
- shift 42,56, and 90 over, then insert 15 at index 1 (93% correct)
- insert 15 at index 1

Warmup #2

Considering your answer to the warmup question 1, insertion at what location requires the *least* operations?

- at beginning
- at the end (80% correct)
- in the middle
- there is no preference

Warmup #3

Considering your answer to the warmup question 1, insertion at what location requires the *most* operations?

- at beginning (73% correct)
- at the end
- in the middle
- there is no preference

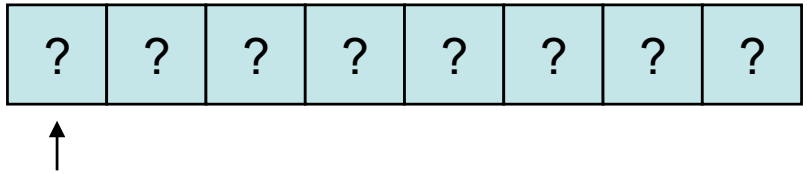
A partial Ordered List ADT

List with operations:

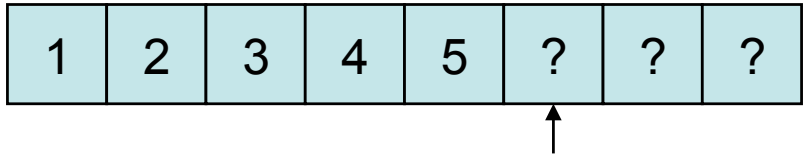
- insert(size_t index, TYPE value), expanding the list
- remove(size_t index), remove value from list, compress the list

Insertion cases using arrays

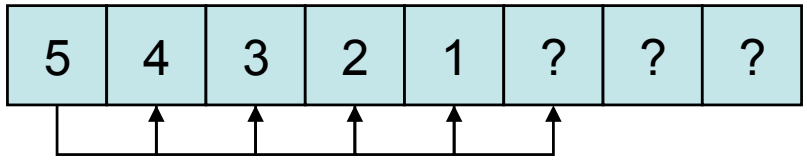
What happens on the first insertion? (size = 0)



What happens if the insertion is always at the end?

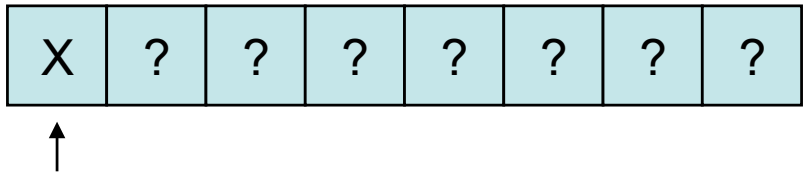


What happens if the insertion is always at the beginning?

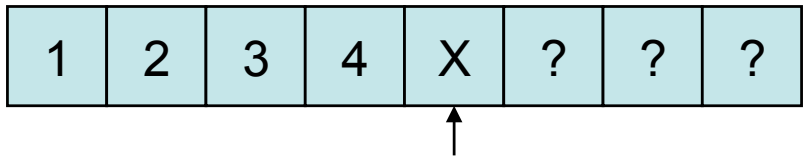


Deletion cases using arrays

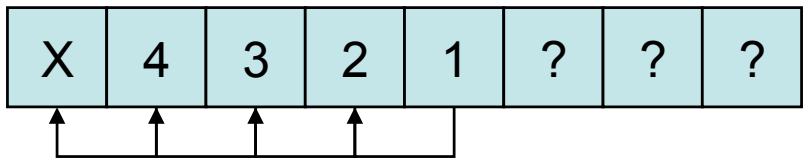
What happens on the last deletion?



What happens if the deletion is always at the end?



What happens if the deletion is always at the beginning?



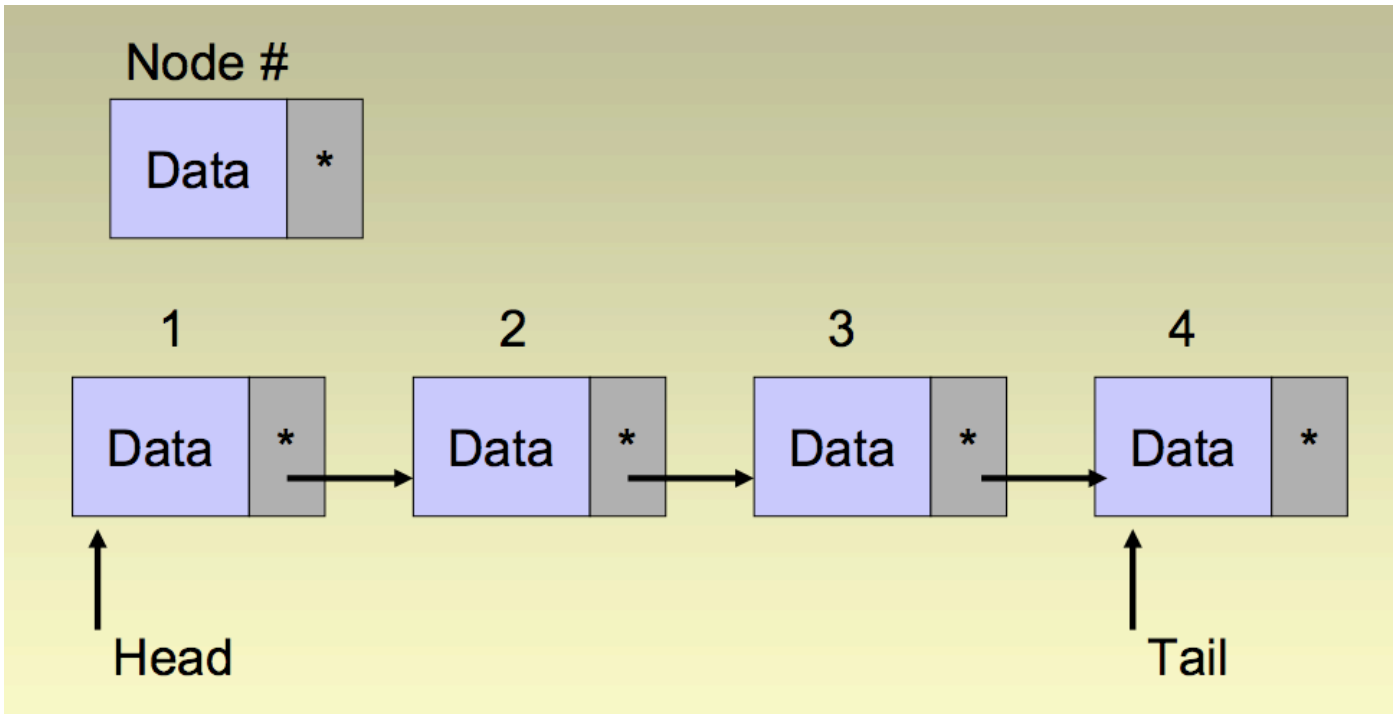
The implementation of the ADT Bag and Ordered List using arrays can be replaced with another data structure.

A **linked list** is a data structure that has several advantages over the array data structure:

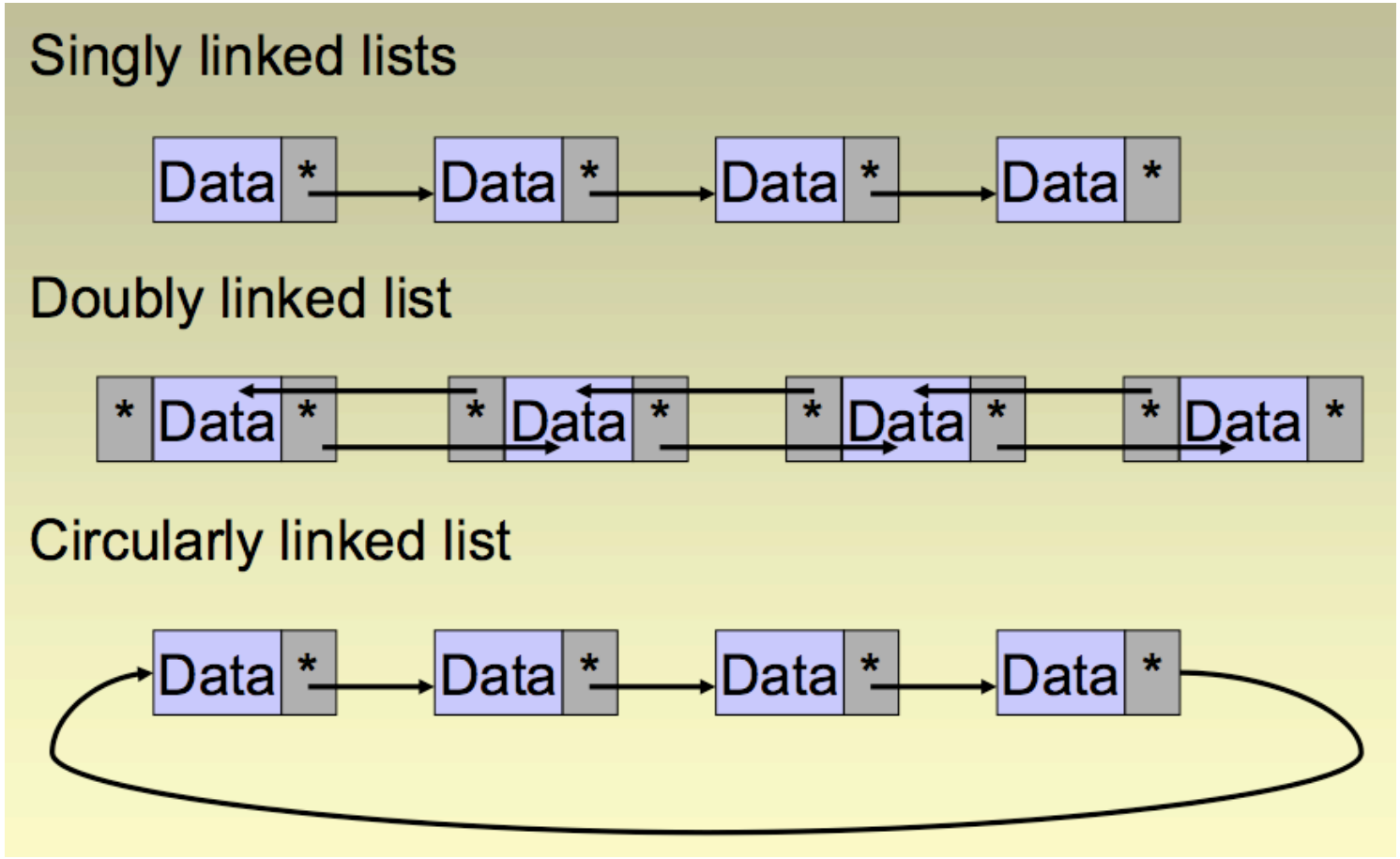
1. It can grow and shrink as needed without wasting space
2. Insertions and deletions can be done with no copying

The disadvantage compared to an array-based implementation is the retrieve operations are no longer constant in the general case. It is constant to retrieve the beginning and end item though.

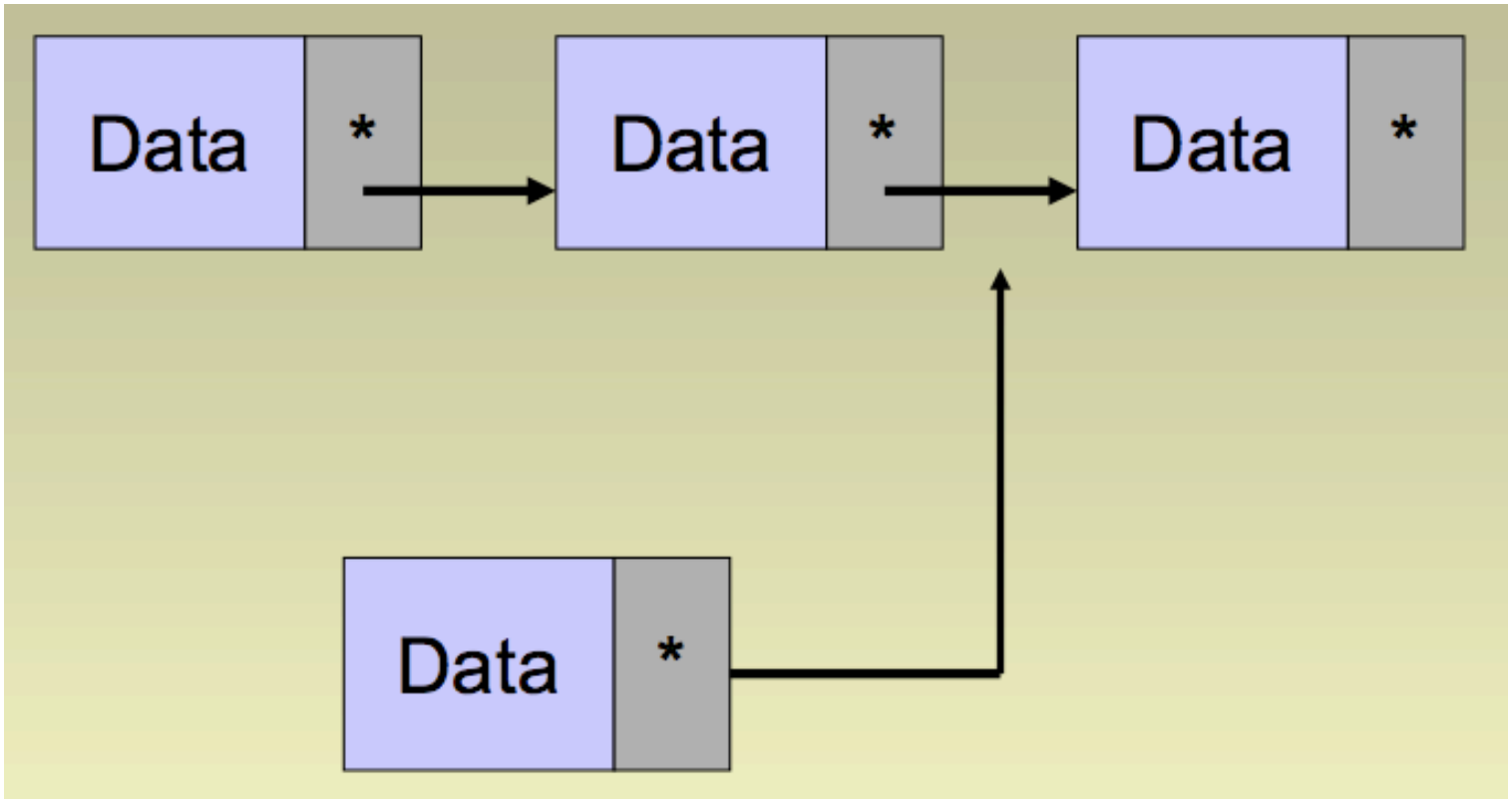
The linked list data structure is a set of nodes linked by pointers.



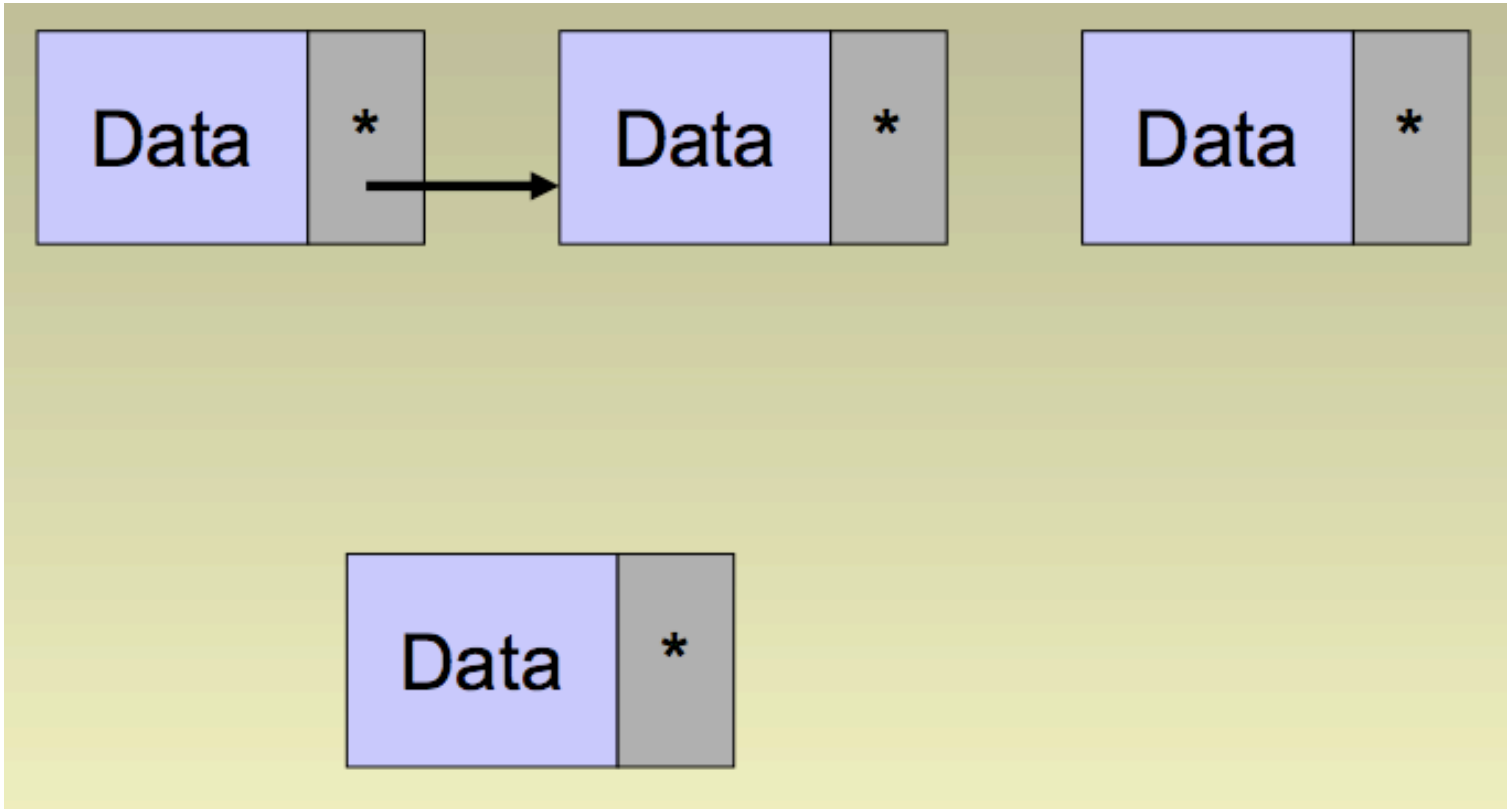
There are three main variations on the linked list.



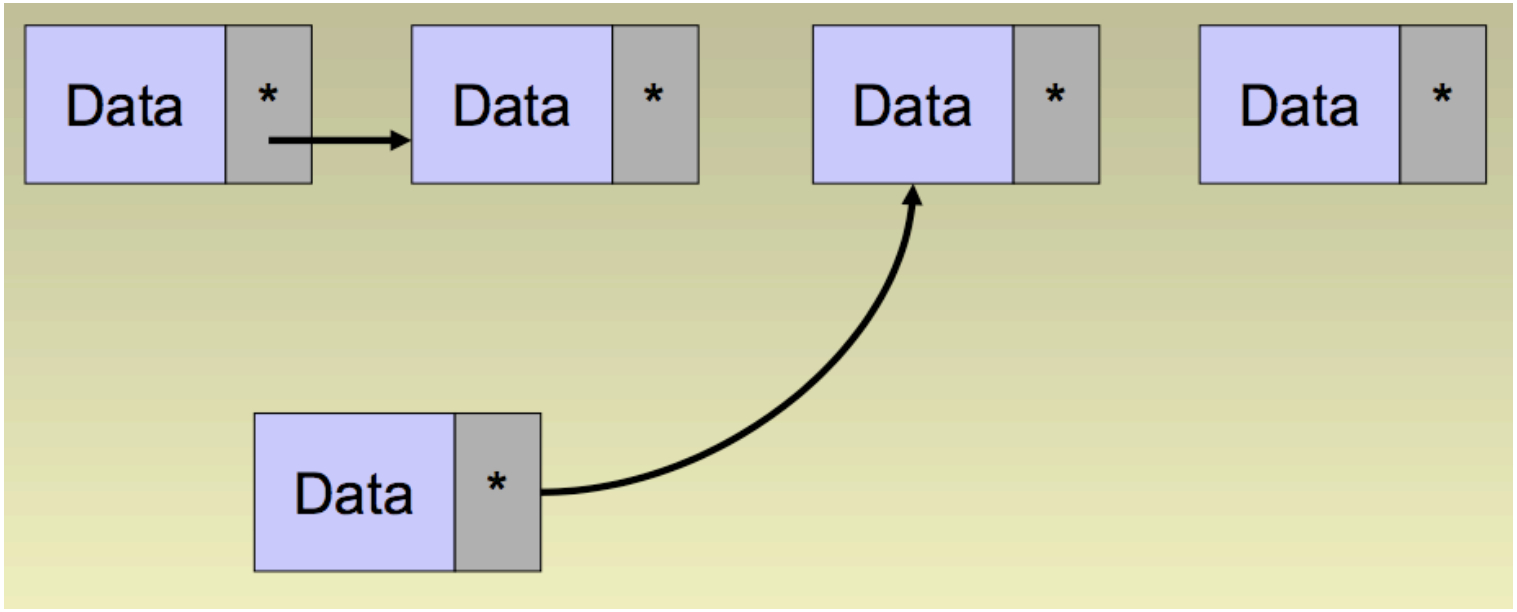
Overview of the insert operation



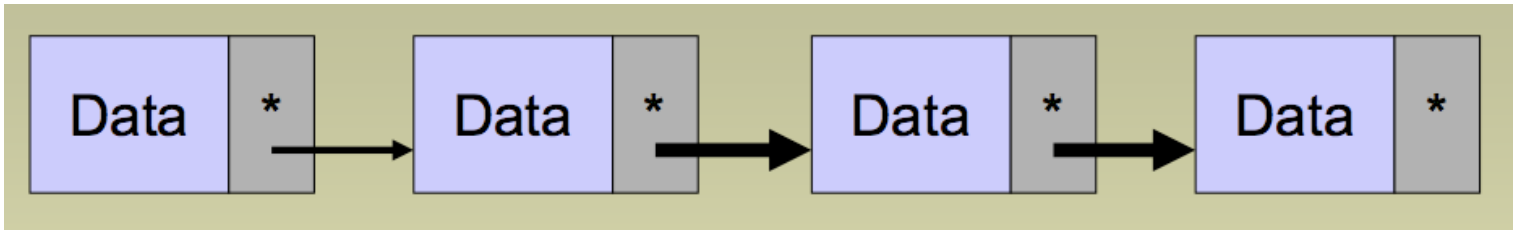
Overview of the insert operation



Overview of the insert operation

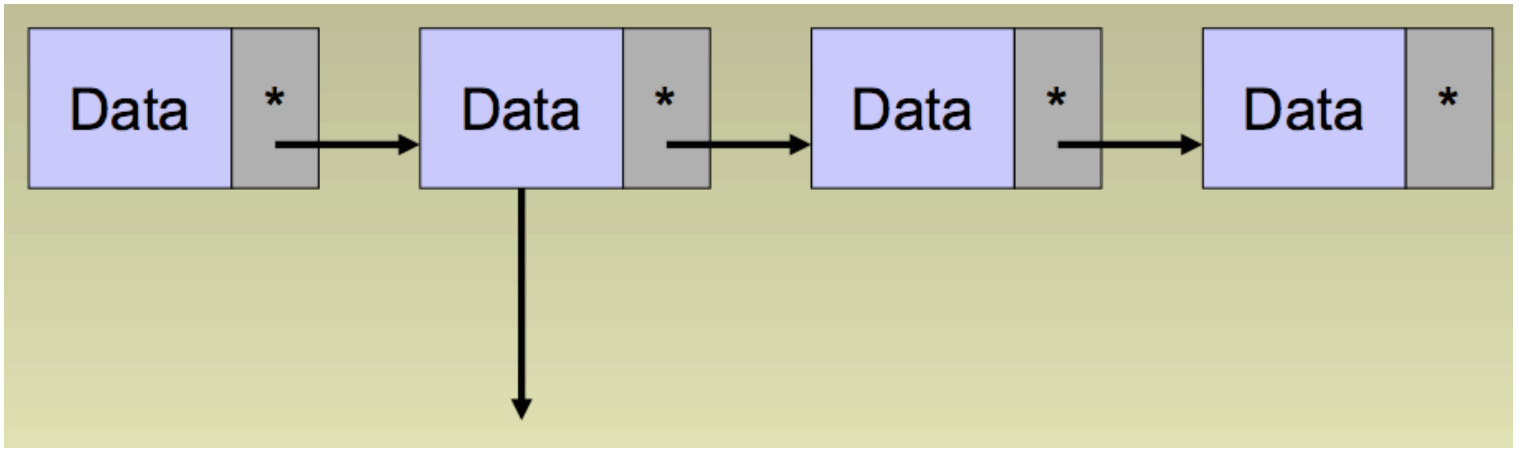


Overview of the insert operation

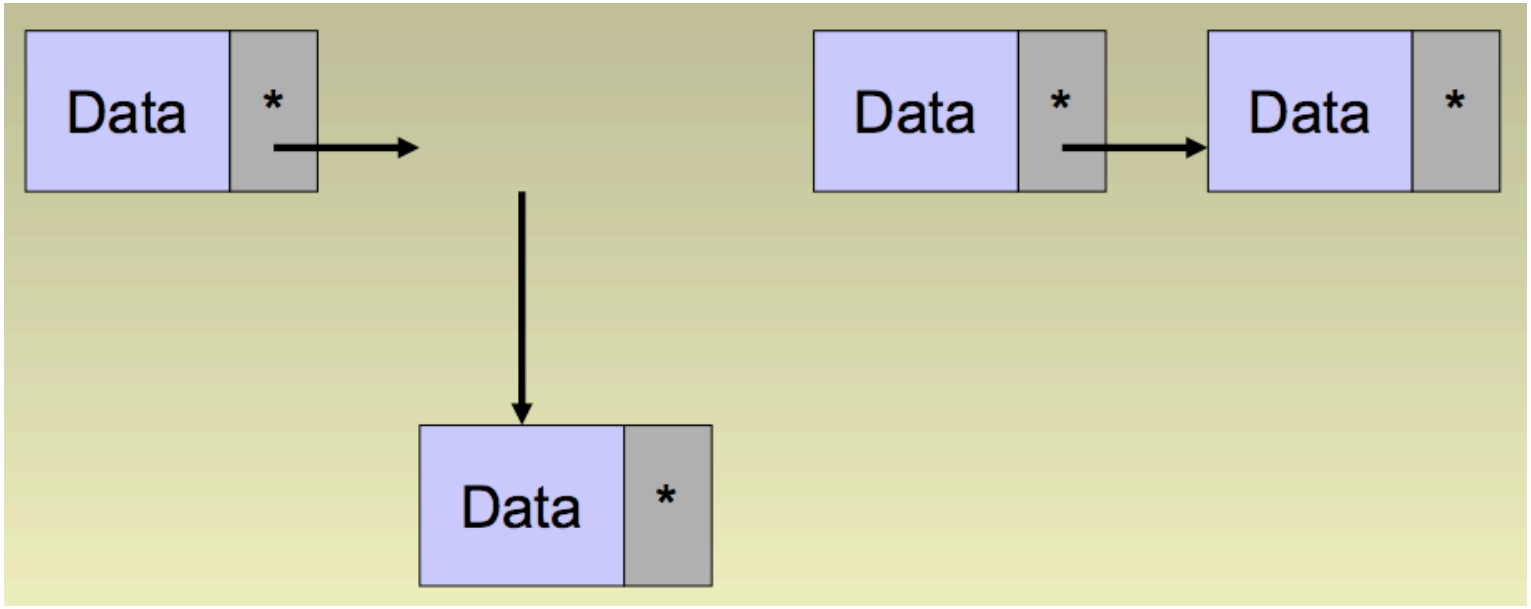


Notice only the amount of memory needed to hold the item was allocated and no copying was required.

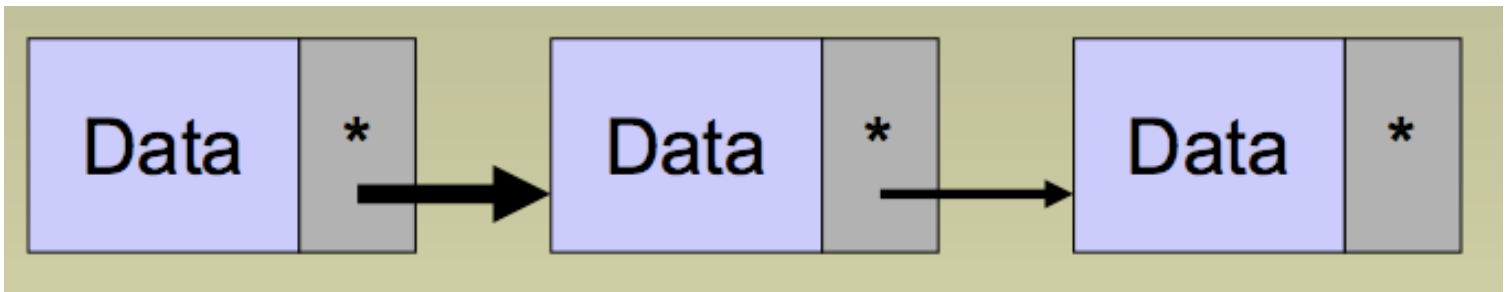
Overview of the delete operation



Overview of the delete operation



Overview of the delete operation



Again, notice, no copying was required

Implementing the single linked list

Need to represent nodes inside the class.

Example: (see in-class code, no exercise today)

- Define a single-linked Bag template class using a structure to represent the nodes.
- Implement the basic operations.

Next Actions and Reminders

- Read CH pp. 150-154
- No warmup for next time.
- P1 is due 9/27