

Doubly linked
list

Definitions

Differences with
singly

Implementation

Insertion

Removal

Sketchpad
slide

Comparison

Table of step counts

Advantages and
disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

Doubly linked lists and freeList node caches

Comp Sci 1575 Data Structures



Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

The code that is the hardest to debug is the code you know can't possibly be wrong.

Doubly linked list

- Definitions
- Differences with singly
- Implementation
- Insertion
- Removal

Sketchpad slide

Comparison

- Table of step counts
- Advantages and disadvantages
- Variations

Freelist

- Problem
- Solution
- Caching
- Shared freelists
- Efficiency
- Code

Do Q1

Doubly linked list

- Definitions
- Differences with singly
- Implementation
- Insertion
- Removal

Sketchpad slide

Comparison

- Table of step counts
- Advantages and disadvantages
- Variations

Freelist

- Problem
- Solution
- Caching
- Shared freelists
- Efficiency
- Code

1 Doubly linked list

- Definitions
- Differences with singly
- Implementation
- Insertion
- Removal

2 Sketchpad slide

3 Comparison

- Table of step counts
- Advantages and disadvantages
- Variations

4 Freelist

- Problem
- Solution
- Caching
- Shared freelists
- Efficiency
- Code

Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

1 Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

2 Sketchpad slide

3 Comparison

Table of step counts

Advantages and disadvantages

Variations

4 Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

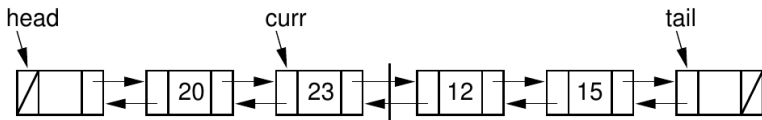
Solution

Caching

Shared freelists

Efficiency

Code



- Doubly linked list has a set of sequentially linked nodes.
- Each node contains pointers to the previous and next node.
- Head and tail nodes are often empty/null, but not always.

Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

1 Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

2 Sketchpad slide

3 Comparison

Table of step counts

Advantages and disadvantages

Variations

4 Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

Doubly linked lists difference with singly linked

Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

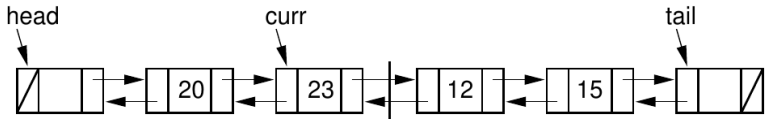
Solution

Caching

Shared freelists

Efficiency

Code



- We no longer need to insert after current; however to keep the same functions as we have for the singly linked list, we will keep curr as pointing to the item before the operations being performed.
- Some functions, like `prev()`, are simpler with doubly linked lists than singly

Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

1 Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

2 Sketchpad slide

3 Comparison

Table of step counts

Advantages and disadvantages

Variations

4 Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

Very few changes to node class from singly linked

Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

Comparing *node_FL.h* to *node_FL_DL.h* :

- Add second pointer, pointing to previous node
- Each node now has two pointers and a data element
- Tweak constructor slightly to accommodate new pointer
- Check it out!

Doubly linked list

- Definitions
- Differences with singly

Implementation

- Insertion
- Removal

Sketchpad slide

Comparison

- Table of step counts
- Advantages and disadvantages
- Variations

Freelist

- Problem
- Solution
- Caching
- Shared freelists
- Efficiency
- Code

Do Q2

Very few changes to doubly from singly linked class

Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

Comparing *list_L.h* to *list_DL.h* :

- `insert()`, `append()`, `remove()`, and `prev()` are the only functions which change appreciably.
- Check it out!

Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

1 Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

2 Sketchpad slide

3 Comparison

Table of step counts

Advantages and disadvantages

Variations

4 Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

Doubly linked list

- Definitions
- Differences with singly
- Implementation
- Insertion**
- Removal

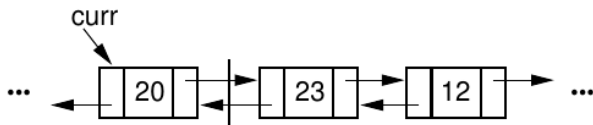
Sketchpad slide

Comparison

- Table of step counts
- Advantages and disadvantages
- Variations

Freelist

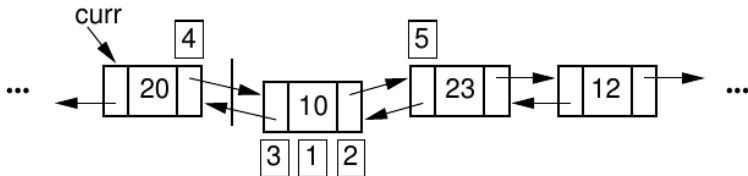
- Problem
- Solution
- Caching
- Shared freelists
- Efficiency
- Code



Insert 10:

	10	
--	----	--

(a)



Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

1 Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

2 Sketchpad slide

3 Comparison

Table of step counts

Advantages and disadvantages

Variations

4 Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

Doubly linked list

- Definitions
- Differences with singly
- Implementation
- Insertion
- Removal**

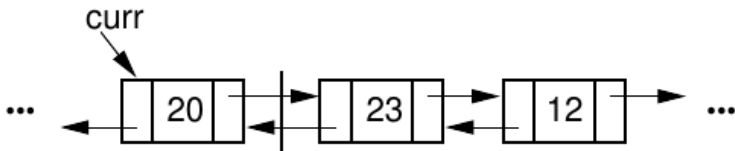
Sketchpad slide

Comparison

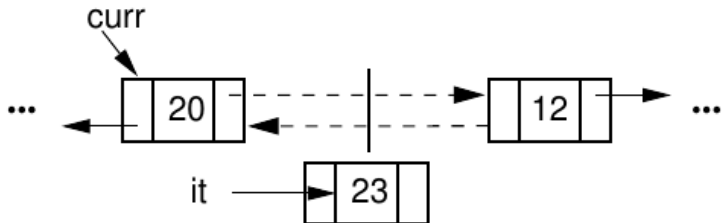
- Table of step counts
- Advantages and disadvantages
- Variations

Freelist

- Problem
- Solution
- Caching
- Shared freelists
- Efficiency
- Code



(a)



Doubly linked list

- Definitions
- Differences with singly
- Implementation
- Insertion
- Removal

Sketchpad slide

Comparison

- Table of step counts
- Advantages and disadvantages
- Variations

Freelist

- Problem
- Solution
- Caching
- Shared freelists
- Efficiency
- Code

- ### 1 Doubly linked list

 - Definitions
 - Differences with singly
 - Implementation
 - Insertion
 - Removal
- ### 2 Sketchpad slide
- ### 3 Comparison

 - Table of step counts
 - Advantages and disadvantages
 - Variations
- ### 4 Freelist

 - Problem
 - Solution
 - Caching
 - Shared freelists
 - Efficiency
 - Code

Doubly linked list

- Definitions
- Differences with singly
- Implementation
- Insertion
- Removal

Sketchpad slide

Comparison

- Table of step counts
- Advantages and disadvantages
- Variations

Freelist

- Problem
- Solution
- Caching
- Shared freelists
- Efficiency
- Code

Doubly linked list

- Definitions
- Differences with singly
- Implementation
- Insertion
- Removal

Sketchpad slide

Comparison

- Table of step counts
- Advantages and disadvantages
- Variations

Freelist

- Problem
- Solution
- Caching
- Shared freelists
- Efficiency
- Code

Do code

Doubly linked list

- Definitions
- Differences with singly
- Implementation
- Insertion
- Removal

Sketchpad slide

Comparison

- Table of step counts
- Advantages and disadvantages
- Variations

Freelist

- Problem
- Solution
- Caching
- Shared freelists
- Efficiency
- Code

Do Q3

Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

- 1 **Doubly linked list**
 - Definitions
 - Differences with singly
 - Implementation
 - Insertion
 - Removal

- 2 **Sketchpad slide**

- 3 **Comparison**
 - Table of step counts
 - Advantages and disadvantages
 - Variations

- 4 **Freelist**
 - Problem
 - Solution
 - Caching
 - Shared freelists
 - Efficiency
 - Code

Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

- 1 **Doubly linked list**
 - Definitions
 - Differences with singly
 - Implementation
 - Insertion
 - Removal

- 2 **Sketchpad slide**

- 3 **Comparison**
 - Table of step counts**
 - Advantages and disadvantages
 - Variations

- 4 **Freelist**
 - Problem
 - Solution
 - Caching
 - Shared freelists
 - Efficiency
 - Code

Operation steps related list size increase?

Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

	ArrayList	LinkedList	DLinkedList
Insert()	n	1	1
append()	1	1	1
remove()	n	1	1
moveToStart/End()	1	1	1
prev()	1	n	1
next()	1	1	1
length() cnt or re-calc	1	1 or n	1 or n
currPos()	1	n or 1	n or 1
moveToPos()	1	n	n
getValue() curr or spec	1	1 or n	1 or n
clear()	=	≥	≥

Differences between these data structures are moderate, but for other structures, choosing wrong might be the difference between tractable and not

Doubly linked list

- Definitions
- Differences with singly
- Implementation
- Insertion
- Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

- Problem
- Solution
- Caching
- Shared freelists
- Efficiency
- Code

- 1 **Doubly linked list**
 - Definitions
 - Differences with singly
 - Implementation
 - Insertion
 - Removal

- 2 **Sketchpad slide**

- 3 **Comparison**
 - Table of step counts
 - Advantages and disadvantages**
 - Variations

- 4 **Freelist**
 - Problem
 - Solution
 - Caching
 - Shared freelists
 - Efficiency
 - Code

Advantages and disadvantages

Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

- Only significant disadvantage of doubly linked list compared to singly is additional space usage
- Only significant advantage over LinkedList is `prev()`

Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

- 1 **Doubly linked list**
 - Definitions
 - Differences with singly
 - Implementation
 - Insertion
 - Removal

- 2 **Sketchpad slide**

- 3 **Comparison**
 - Table of step counts
 - Advantages and disadvantages
 - Variations**

- 4 **Freelist**
 - Problem
 - Solution
 - Caching
 - Shared freelists
 - Efficiency
 - Code

Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

- Circularly linked lists (instead of sentinel head/tail nodes)
- Data elements which are just pointers, so multiple lists can point to the same data: the larger the elements and the more they are duplicated, the more likely that pointers to shared elements is the better approach.

Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

- 1 **Doubly linked list**
 - Definitions
 - Differences with singly
 - Implementation
 - Insertion
 - Removal
- 2 **Sketchpad slide**
- 3 **Comparison**
 - Table of step counts
 - Advantages and disadvantages
 - Variations
- 4 **Freelist**
 - Problem
 - Solution
 - Caching
 - Shared freelists
 - Efficiency
 - Code

Doubly linked list

- Definitions
- Differences with singly
- Implementation
- Insertion
- Removal

Sketchpad slide

Comparison

- Table of step counts
- Advantages and disadvantages
- Variations

Freelist

- Problem**
- Solution
- Caching
- Shared freelists
- Efficiency
- Code

- 1 **Doubly linked list**
 - Definitions
 - Differences with singly
 - Implementation
 - Insertion
 - Removal
- 2 **Sketchpad slide**
- 3 **Comparison**
 - Table of step counts
 - Advantages and disadvantages
 - Variations
- 4 **Freelist**
 - Problem**
 - Solution
 - Caching
 - Shared freelists
 - Efficiency
 - Code

Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

- We create and delete nodes regularly with linked lists
- **new** and **delete** are relatively expensive
- With a dynamic array, we might only create a larger array every now and then.
- Is there a similar solution for our linked lists composed of node elements?

Doubly linked list

- Definitions
- Differences with singly
- Implementation
- Insertion
- Removal

Sketchpad slide

Comparison

- Table of step counts
- Advantages and disadvantages
- Variations

Freelist

- Problem
- Solution**
- Caching
- Shared freelists
- Efficiency
- Code

- 1 **Doubly linked list**
 - Definitions
 - Differences with singly
 - Implementation
 - Insertion
 - Removal
- 2 **Sketchpad slide**
- 3 **Comparison**
 - Table of step counts
 - Advantages and disadvantages
 - Variations
- 4 **Freelist**
 - Problem
 - Solution**
 - Caching
 - Shared freelists
 - Efficiency
 - Code

Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

- Generally, cache stores data in an easily accessible temporary location so future requests for that data can be faster (many applications and levels of caching)
- **freelist** holds some nodes not currently being used
- Rather than call standard new and delete:
 - overloaded **new** will take from the node cache and give to the list
 - overloaded **delete** will take from the list and give to the cache

Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

- To **delete** a node from linked list, place at head of the freelist
- To add a **new** node to a linked list, check freelist for available nodes, and if available, take node from head of freelist
- If freelist is empty, standard **new** operator called
- freelist should be a static variable. Why?

Shared freelist between lists of the same type

Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

In the node class itself:

```
static Node<E> *freelist;
```

- This creates a single variable shared among all instances of the Link nodes of a given type
- Even with one template class, the compiler automatically gives each type of the class its own class type, and thus its own freelist

Doubly linked list

Definitions

Differences with singly

Implementation

Insertion

Removal

Sketchpad slide

Comparison

Table of step counts

Advantages and disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

- Are freelists useful if your list only grows and never shrinks?
- Useful for linked lists that periodically grow and then shrink.
- Will never grow larger than largest size yet reached
- Operator overloading of **new** and **delete** is invisible to user, and hidden in node class

Only change: overloaded operators in Node class

Doubly linked
list

Definitions

Differences with
singly

Implementation

Insertion

Removal

Sketchpad
slide

Comparison

Table of step counts

Advantages and
disadvantages

Variations

Freelist

Problem

Solution

Caching

Shared freelists

Efficiency

Code

- Linked list has two classes: *list_L.h* and *node.h*
- *node_FL.h* replaces *node.h*
- The list class (*list_L.h*) doesn't have to change at all, or know that the node class has changed!
- Main just needs a new include *#include "node_FL.h"*
- Check it out.