David Croft

Allays

Linked lists

LL example

structures

types

Queue:

Stacks

Stacks

...

Othe

Tree

Quiz

Recap

122com Data structures and types

David Croft

Coventry University david.croft@coventry.ac.uk

2017



Overview

- 1 Arrays
- 2 Linked lists
 - Array example
 - LL example
- 3 Data structures
- 4 Abstract data types
- 5 Queues
- 6 Stacks
- 7 Sets
- 8 Other
- 9 Trees
- 10 Quiz
- 11 Recap

David Croft

Arrays

Linked list

Array example

LL example

structures
Abstract dat

Abstract data types

Queues

Stacks

Othe

Tree

Quiz

Recap

A series of objects all of the same size and type.

- Stored in contiguous blocks of memory.
- Python lists are functionally closest.
 - But are not arrays.
- Can't be resized.



David Croft

Linked lists

The challenger for array's crown.

- Series of nodes, each of which points to the next element.
 - And to the previous element if it's a doubly linked list.

Doubly linked
$$\leftarrow A \leftarrow B \leftarrow C \rightarrow D$$

$$\leftarrow \mid \mathsf{A} \mid$$

$$| \rightarrow |$$
 \leftarrow

$$| \rightarrow |$$

$$\leftarrow$$





Linked lists

Array example

Data structure

Abstract data

Queues

Stack

Sets

Othe

Tree

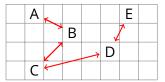
Quiz

Recap



Not in contiguous memory.

- Each node is separate.
- Scattered.
- C++ Dynamic memory (pointers!).
 - Discussed in depth later in module.



- Why would we use linked lists instead of arrays?
 - Can change size.
 - Can quickly insert and delete elements.

```
class Node:
    __prev = None
    __next = None
    value = None
```

Linked lists II

```
class Node
{
private:
    Node *prev;
    Node *next;

public:
    int value;
};
```



Arrays Linked lists

Array example

Data

Abstract data

Queues

Stack

Otho

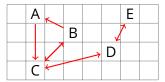
Tree

Recar



Not in contiguous memory.

- Each node is separate.
- Scattered.
- C++ Dynamic memory (pointers!).
 - Discussed in depth later in module.



- Why would we use linked lists instead of arrays?
 - Can change size.
 - Can quickly insert and delete elements.

```
class Node:
    __prev = None
    __next = None
    value = None
```

Linked lists II

```
class Node
{
private:
    Node *prev;
    Node *next;

public:
    int value;
};
```

Linked lists

Data

Abstract data

Queues

Queues

Stack

Otho

Tvoo

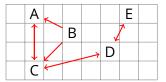
1100

Recar



Not in contiguous memory.

- Each node is separate.
- Scattered.
- C++ Dynamic memory (pointers!).
 - Discussed in depth later in module.



- Why would we use linked lists instead of arrays?
 - Can change size.
 - Can quickly insert and delete elements.

```
class Node:
    __prev = None
    __next = None
    value = None
```

Linked lists II

```
class Node
{
private:
    Node *prev;
    Node *next;

public:
    int value;
};
```



Abstract da types

Queues

Stack

Sets

Othe

Tree

.

Recap

Coventry University



```
temp = ['A', 'B', 'C', 'D', 'E']
array<char,5> temp {'A', 'B', 'C', 'D', 'E'};
```



- Array in memory, multiple elements in a contiguous block.
- How do we remove elements from the middle?

Abstract da types

Queue

Ctack

_ .

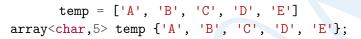
Othe

_

...

Recap

Coventry University





- Array in memory, multiple elements in a contiguous block.
- How do we remove elements from the middle?
 - Remove element from the array.

Abstract dat types

Queue

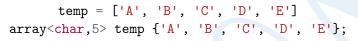
c. .

Sets

Othe

Tree

Recap





- Array in memory, multiple elements in a contiguous block.
- How do we remove elements from the middle?
 - Remove element from the array.
 - Move next element to occupy the empty space.



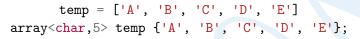
Coventry University

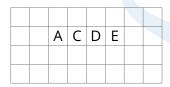




- Array in memory, multiple elements in a contiguous block.
- How do we remove elements from the middle?
 - Remove element from the array.
 - Move next element to occupy the empty space.
 - Repeat.

Coventry University





- Array in memory, multiple elements in a contiguous block.
- How do we remove elements from the middle?
 - Remove element from the array.
 - Move next element to occupy the empty space.
 - Repeat.



- Array in memory, multiple elements in a contiguous block.
- How do we remove elements from the middle?
 - Remove element from the array.
 - Move next element to occupy the empty space.
 - Repeat.
- Is very slow with large arrays.

Abstract dat

types

Queue:

Stack

Sets

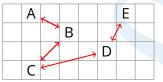
Othe

Troo

O...:-

Recap





- Linked list, separate elements scattered in memory.
- Each pointing to the next/prev element.
- How do we remove elements?

Abstract dat

-71---

Queue

Stack

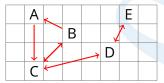
Sets

Othe

Trac

Recap





- Linked list, separate elements scattered in memory.
- Each pointing to the next/prev element.
- How do we remove elements?
 - Change pointers.

Queue

Stack

Jeach

Sets

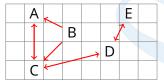
Othe

Tree

Quiz

Recap





- Linked list, separate elements scattered in memory.
- Each pointing to the next/prev element.
- How do we remove elements?
 - Change pointers.

Coventry University



- Linked list, separate elements scattered in memory.
- Each pointing to the next/prev element.
- How do we remove elements?
 - Change pointers.
 - Delete old element.

David Croft

Linked lis

Data structure

Abstract dat

Queues

Stack

Sets

Othe

Tree

1100

Recar

Coventry University

Advantages

- Inserting and deleting elements is very fast.
 - O(1).
- No size limits, can keep adding new elements.
- Doesn't waste memory.

Disadvantages

- Not indexed.
 - Can't ask for the 20th element etc.
 - Have to step through the list (slow).
- Needs more memory than an array to store the same number of elements.
 - Have to keep track of where the next/prev nodes are.

Array example

Data structures

Abstract types

Ouelles

Stack

Sets

Othe

Tree

Quiz

Recap

Arrays and linked lists are data structures.

- A specific way of storing data.
- Can see how the various elements of the structure are laid out in memory.
- Direct access to the underlying memory.



Array example

Abstract data

Abstract dat types

Queues

Stack

Sets

Othe

Trac

Quiz

Recan

As we move to storing more complex information in our software we well start to encounter Abstract Data Types (ADTs).

Software engineering principal.



Abstract data types

structures
Abstract data

types

Queue:

Stack

Sets

Otne

Tree

Recap

- Software engineering principal.
- Keep what a data type can do...



Array example

Abstract data

types

Queue:

Stack

J. C. C. C.

O+l- .

O Ci ii

Tree

Recap

- Software engineering principal.
- Keep what a data type can do... ...and how it does it separate.



Linked list
Array example
LL example

structures
Abstract data

Abstract dat types

Oueue:

Stack

Jeach

Oth

Troo

....

Recar

- Software engineering principal.
- Keep what a data type can do... ...and how it does it separate.
- Unlike data structure ADTs only concerned with the interface.



Linked list
Array example
LL example

Abstract data

types

Queues

Stack:

Oth

Tree

<u>۔۔۔۔</u>

Recan

- Software engineering principal.
- Keep what a data type can do... ...and how it does it separate.
- Unlike data structure ADTs only concerned with the interface.
- Internals of ADTs can vary widely between implementations.



122com Data

Abstract data types

Queue

Stack

Othe

Two o

Quiz

Recap

Imagine an ADT like a car.

It has a set of supported operations, go faster, go slower, turn left, turn right.





types

David Croft

122com Data structs &

Arrays

Linked lists
Array example
LL example

Abstract data

types

Queue:

Stack:

Sets

Othe

Tree

Quiz

Recap

Imagine an ADT like a car.

- It has a set of supported operations, go faster, go slower, turn left, turn right.
- Don't care how it achieves these.



David Croft

Arrays

Linked lists
Array example
LL example

Abstract data

types

Queue

Stack

Recap

Imagine an ADT like a car.

- It has a set of supported operations, go faster, go slower, turn left, turn right.
- Don't care how it achieves these.
- Don't care if, internally, it's using a combustion engine or an electric motor.



Linked lists Array example LL example

Abstract data

types

Queue:

Staci

sets

Othe

Tree

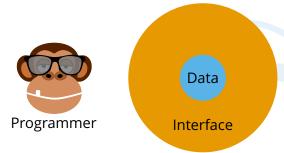
Qui

Recap

LL example Data structures

Imagine an ADT like a car.

- It has a set of supported operations, go faster, go slower, turn left, turn right.
- Don't care how it achieves these.
- Don't care if, internally, it's using a combustion engine or an electric motor.
- Only care about the result.
- Keep people away from the internal workings/data.





Queues

Arrays

Linked lists
Array example
LL example

Abstract data

types

Queues

Stack

Sets

Othe

Tree

~ ----

Recap

A First In First Out (FIFO) ADT.

- Ends of the queue called the front and back.
- New elements added to back of queue only.
 - Pushing push(value)
- Old elements removed from front of queue only.
 - Popping pop()
- No cutting in.



David Croft

Arrays

Linked lists
Array example
LL example

Abstract data

cypes

Queues

Stacks

sets

Othe

Tree

Quiz

Recap

A FIFO ADT.

- Ends of the queue called the front and back.
- New elements added to back of queue only.
 - Pushing push(value)
- Old elements removed from front of queue only.
 - Popping pop()
- No cutting in.
- Buffer to hold items for processing in the order in which they arrive.



Queues

Coventry University

A FIFO ADT.

- Ends of the queue called the front and back.
- New elements added to back of queue only.
 - Pushing push(value)
- Old elements removed from front of queue only.
 - Popping pop()
- No cutting in.
- Buffer to hold items for processing in the order in which they arrive.
- Which would be better for a queue? An array or a linked list?

Queues

Coventry University

A FIFO ADT.

- Ends of the queue called the front and back.
- New elements added to back of queue only.
 - Pushing push(value)
- Old elements removed from front of queue only.
 - Popping pop()
- No cutting in.
- Buffer to hold items for processing in the order in which they arrive.
- Which would be better for a queue? An array or a linked list?
 - Linked list.

 $front \Rightarrow$

types

Queues

Stack

Otilie

Ouiz

Recap



Array as a queue.

- Very similar to stacks.
 - Keep track of next free space.
 - Limited size.

Queues

Queue.

Stack

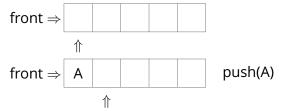
...

Quiz

Recap







- Very similar to stacks.
 - Keep track of next free space.
 - Limited size.



Queues

Stack

sets

0

Quiz

Recap

Array as a queue.

- front ⇒ ↑
- front \Rightarrow A

push(A)

 \uparrow

front \Rightarrow A B

push(B)



- Very similar to stacks.
 - Keep track of next free space.
 - Limited size.



Abstract data

Queues

Stacks

. .

Othe

Troo

Ouiz

Recap

Array as a queue.

I

- front ⇒ ↑
- front $\Rightarrow A$

push(A)

 \uparrow

front \Rightarrow A B

push(B)

 \uparrow

front \Rightarrow A B C

push(C)



- Very similar to stacks.
 - Keep track of next free space.
 - Limited size.



Linked list

Data structures

Abstract data types

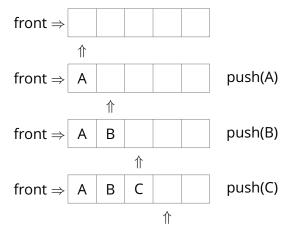
Queues

C+--l--

_ ----

Othe

Traa



- Very similar to stacks.
 - Keep track of next free space.
 - Limited size.
- What happens when we pop()?
 - Have to shuffle every element forward one space.
 - Inefficient.



Linked list: Array example

Data structures

Abstract data types

Queues

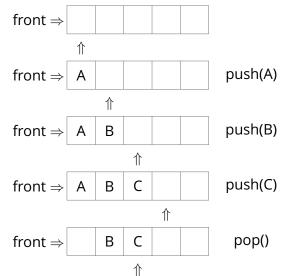
Ctooko

Coto

Othe

Tree

Ouiz



- Very similar to stacks.
 - Keep track of next free space.
 - Limited size.
- What happens when we pop()?
 - Have to shuffle every element forward one space.
 - Inefficient.



Array as a queue.

1

A ****

Linked lists Array example

Data

Abstract data

Queues

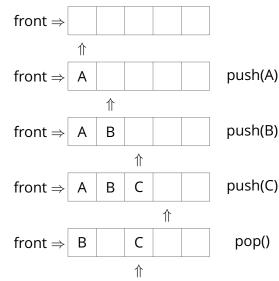
Stacks

Soto

Othe

Tree

Ouiz



- Very similar to stacks.
 - Keep track of next free space.
 - Limited size.
- What happens when we pop()?
 - Have to shuffle every element forward one space.
 - Inefficient.



Abstract data

Queues

_

Otho

Troo

Recap



Array as a queue.

- front \Rightarrow \uparrow
- front \Rightarrow A
 - \uparrow
- front \Rightarrow A B push(B)
- front \Rightarrow A B C push(C)
- front \Rightarrow B C \uparrow
- pop()

push(A)

- Very similar to stacks.
 - Keep track of next free space.
 - Limited size.
- What happens when we pop()?
 - Have to shuffle every element forward one space.
 - Inefficient.

Data

Abstract dat

Queues

Stack:

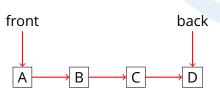
Othe

Tree

Quiz

Recap

Linked list as a queue.





Linked list

LL example

structures

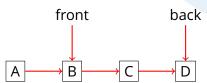
pop()

Abstract data types

Queues

. .

Ouiz





Linked list

LL example

structures

pop()

Abstract data types

Queues

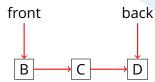
Stacks

. .

Otho

Troo

Quiz





Linked list

LL example

structures

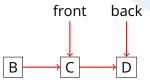
Abstract data types

Queues

Stacks

Recap

pop(), pop()





Abstract data types

Queues

Stacks

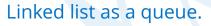
Sets

Othe

Tree

Quiz

Recap





pop(), pop()





Stack:

Othe

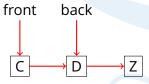
Tree

Quiz

Recap



pop(), pop(), push(Z)





Queues

Stack

0.1

Othic

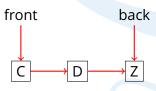
....

Recap





pop(), pop(), push(Z)





Stacks

Quiz

Recap

A First In Last Out (FILO) ADT.

- Ends of the stack are called the top and bottom.
- New elements add to top of stack only.
 - Pushing push(value)
- Old elements removed from top of stack only.
 - Popping pop()
- No cutting in.



David Croft

Array

Linked list Array example LL example

Structures
Abstract data

types

Queues

Stacks

Jeach

Othe

Traa

Recap

A FILO ADT.

- Ends of the stack are called the top and bottom.
- New elements add to top of stack only.
 - Pushing push(value)
- Old elements removed from top of stack only.
 - Popping pop()
- No cutting in.
- Which would be better for a stack? An array or a linked list?



Stacks

A FILO ADT.

- Ends of the stack are called the top and bottom.
- New elements add to top of stack only.
 - Pushing push(value)
- Old elements removed from top of stack only.
 - Popping pop()
- No cutting in.
- Which would be better for a stack? An array or a linked list?
 - Doesn't matter performance wise.
 - Linked list if *n* is unknown.



Abstract dat

21

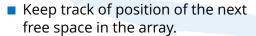
Stacks

Othic

Quiz







- Arrays have a fixed size.
 - Can't hold more values than we have space for.



structures

Abstract data types

Queues

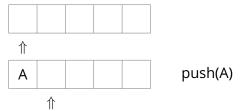
Stacks

Othe

Tree







- Keep track of position of the next free space in the array.
- Arrays have a fixed size.
 - Can't hold more values than we have space for.



Data structure:

Abstract data types

Queues

Queue.

Stacks

5005

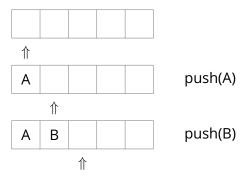
Otne

....

Recan

Array as a stack.





- Keep track of position of the next free space in the array.
- Arrays have a fixed size.
 - Can't hold more values than we have space for.



Arrays

Array example

Abstract dat

types

Queues

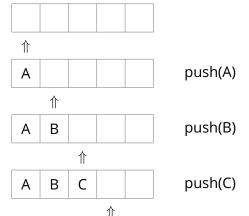
Stacks

Sets

Othe

Tree

Quiz



- Keep track of position of the next free space in the array.
- Arrays have a fixed size.
 - Can't hold more values than we have space for.



Stacks



Array as a stack.

- Keep track of position of the next free space in the array.
- Arrays have a fixed size.
 - Can't hold more values than we have space for.





push(A)





push(B)





push(C)



Α	В		
		⇑	

pop()



Abstract da

Queue

Stack

Sets

Othe

Tree

Pocar



- An unordered ADT.
 - Items ordered by the set.
 - You have no control over it.
- Sets contain unique elements.
 - Can't contain duplicates.
- Can add items to a set.
- Can remove items from a set.
- Can see if an item is in a set.
- Can't get the *n*th element.
 - It's unordered remember.









- An unordered ADT.
 - Items ordered by the set.
 - You have no control over it.
- Sets contain unique elements.
 - Can't contain duplicates.
- Can add items to a set.
- Can remove items from a set.
- Can see if an item is in a set.
- Can't get the *n*th element.
 - It's unordered remember.





add(A)

Array example

structure:

Abstract data types

Queue:

Stack

Sets

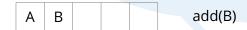
Oth

Tree

- An unordered ADT.
 - Items ordered by the set.
 - You have no control over it.
- Sets contain unique elements.
 - Can't contain duplicates.
- Can add items to a set.
- Can remove items from a set.
- Can see if an item is in a set.
- \blacksquare Can't get the n^{th} element.
 - It's unordered remember.







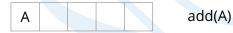


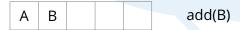
Sets

Sets

- An unordered ADT.
 - Items ordered by the set.
 - You have no control over it.
- Sets contain unique elements.
 - Can't contain duplicates.
- Can add items to a set.
- Can remove items from a set.
- Can see if an item is in a set.
- Can't get the *n*th element.
 - It's unordered remember.











Sets

Sets

- An unordered ADT.
 - Items ordered by the set.
 - You have no control over it.
- Sets contain unique elements.
 - Can't contain duplicates.
- Can add items to a set.
- Can remove items from a set.
- Can see if an item is in a set.
- Can't get the *n*th element.
 - It's unordered remember.















remove(A)

add(A)



...and the others

Arrays

Array example

Data structures

Abstract data

Queues

Stack

Sets

Other

Tree:

Quiz

Recap

Lots of other ADTs.

- Different names in different languages.
- Lists.
- Circular lists.
- Associative arrays.
 - Dictionaries/Maps.
- Double-ended queues.
- Trees.
- Graphs.



Trees

types

Queue

Stack

041-

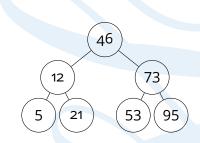
Trees

Ouiz

Recap

Variation on linked lists.

- Made of nodes and relationships.
- Root node at top.
- Each node can have > o children.
- Binary search tree.
 - Very common type.
 - Ordered.
 - Max two children.
 - Binary searching.
 - Very good for sets.





Balance

Queues

Stack

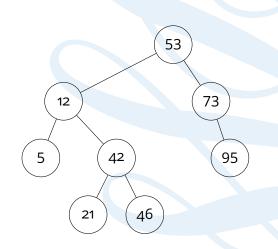
Coto

Othe

Trees

....

- Trees can be balanced or unbalanced.
- Not required for all trees.
- Going to be talking about BSTs from here on.
- Unbalanced because more than a one node difference between the two halves.





Balance

structures

types

Queue:

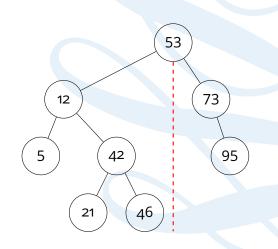
Stack

Soto

Othe

Trees

- Trees can be balanced or unbalanced.
- Not required for all trees.
- Going to be talking about BSTs from here on.
- Unbalanced because more than a one node difference between the two halves.
 - For the whole tree...





Balance

Data structures

Abstract data types

Queue:

Stack

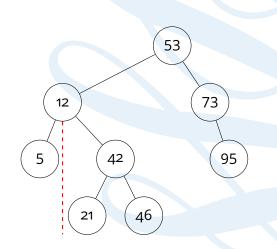
C - 4 -

Otho

Trees

1100

- Trees can be balanced or unbalanced.
- Not required for all trees.
- Going to be talking about BSTs from here on.
- Unbalanced because more than a one node difference between the two halves.
 - For the whole tree...
 - ...and one of the subtrees.



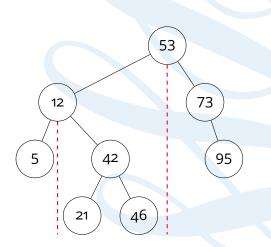


Coventr

Balance



- Trees can be balanced or unbalanced.
- Not required for all trees.
- Going to be talking about BSTs from here on.
- Unbalanced because more than a one node difference between the two halves.
 - For the whole tree...
 - ...and one of the subtrees.





David Croft

Array:

Linked lists
Array example

Data structures

Abstract data

Queues

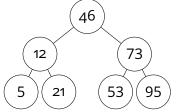
Jeuc.

Trees

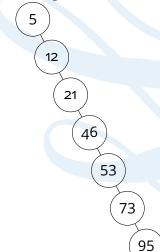
Coventry University

Important that you keep your BSTs balanced.





Degenerate tree.



122com Data structs & types

David Croft

Arrays

Linked lists

Data

Abstract d

Queuc

Stack:

Sets

Otne

Hee

Pocan



Quiz

David Croft

Arrays

Array example

structures

Abstract data types

Queues

Stacks

Stack:

O+l- -

Troo

Quiz

Recap

Stacks and queues are examples of _____

- Data structures.
- Linked lists.
- Arrays.
- Abstract Data Types.



David Croft

Arrays

Array example

Data structure

Abstract data types

Stacks

Sets

Othe

Tree

Quiz

Recap

Stacks and queues are examples of _____

- Data structures.
- Linked lists.
- Arrays.
- Abstract Data Types.



122com Data structs & types David Croft

David Cit

Linked list

Array example LL example

Structures

Abstract dat types

Oueues

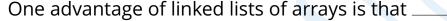
Stacks

Coto

Othe

Tree

Quiz



- They use less memory.
- They don't waste memory.
- They can be used for queues.
- They are faster to search though.



122com Data structs & types David Croft

Arrays

Linked lists
Array example
LL example

structures

Abstract dat types

Queues

Stacks

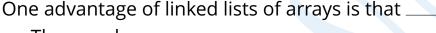
Stacks

Sets

Othe

Tree

Quiz



- They use less memory.
- They don't waste memory.
- They can be used for queues.
- They are faster to search though.



David Croft

Arrays

Array example

Data structure

Abstract data

types

Queue

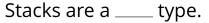
Stack

Soto

Other

Tree

Quiz



- FIFO.
- FOFI.
- FILO.
- FIDO.



David Croft

Allays

Array example

LL example

Structures
Abstract dat

Abstract data types

Queues

Charles

_

....

Recap

Stacks are a ____ type.

- FIFO.
- FOFI.
- FILO.
- FIDO.



122com Data structs & types

David Croft

Arrays

Array example

Data structure

Abstract dat

0.....

Queue

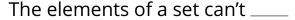
Stack

Sets

Othe

Tree

Quiz



- Contain duplicates.
- Be sequences, ie. lists, strings.
- Be out of order.
- Be removed.



122com Data structs & types

David Croft

Arrays

Array example

Data structur

Abstract data types

Ougues

Queue.

Stack:

_

Quiz

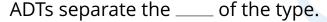
Recap

The elements of a set can't ____

- Contain duplicates.
- Be sequences, ie. lists, strings.
- Be out of order.
- Be removed.



Quiz



- Input and output.
- Attributes and methods.
- Implementation and interface.
- Code and software.



Quiz

ADTs separate the ____ of the type.

- Input and output.
- Attributes and methods.
- Implementation and interface.
- Code and software.



types

Queues

Stack

Jeaci

Otho

....

rree

Quiz

Recap



Everyone

- Need to understand the structures before we can pick the right one.
- Different data structures have very different characteristics.
- Huge effect on efficiency of your code.
- If you pick the right ADT it can save you a lot of code.
 - E.g. why write code to check for duplicates? Use a set and they can't exist.
 - **E**.g. why write code to find the most recent addition to a list, use a stack.



Arraye

Linked list Array example LL example

Abstract dat

Abstract dat types

Queue

Stack

Sets

Tree

Quiz

- Arrays.
 - Advantages/disadvantages.
- Linked lists .
 - Advantages/disadvantages.
 - How to insert/delete.
- Difference between data structure and ADTs.
- Stack.
 - FILO.
 - Using an array as one.
 - Using a LL as one.

- Queue.
 - FIFO.
 - Using an array as one.
 - Using a LL as one.
- Sets.
 - No duplicates.
 - Unordered.
- Trees.
 - Balanced/unbalanced.



122com Data structs & types

David Croft

Arrays

Linked lists

Data

Abstract data types

Queue:

Stacks

Juden.

...

0

Ouiz



