David Croft

Allays

Linked lists

LL example

structures

types

Queue:

Stacks

Juck

...

Othe

Tree

₹есар

122com Data structures and types

David Croft

Coventry University david.croft@coventry.ac.uk

2018



Stacks

Other

Trees

Quiz

Recap



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

Stacks

Sets

Other

Tree

Quiz

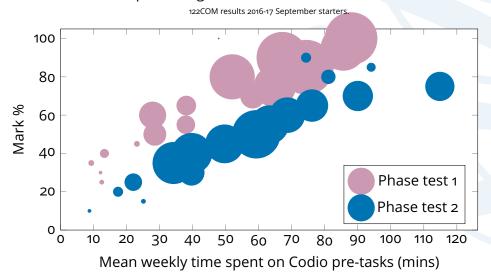
Recap



Expectations



You have all attempted the green Codio exercises for this week.



David Croft

Arrays

Linked list
Array 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

$$\stackrel{\rightarrow}{\leftarrow}$$

$$\stackrel{\rightarrow}{\leftarrow}$$

Arrays Linked lists

Array example

Data structure

Abstract data types

Queues

Stack

Sets

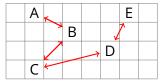
Othe

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
```

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

public:
    int value;
};
```



Linked lists

Array example

Data structure

Abstract data

Queues

Stack

Sets

Othe

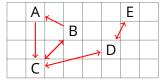
Tree

Quiz



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;

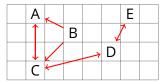
public:
    int value;
};
```





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;
};
```



types

Queues

. .

Sets

Othe

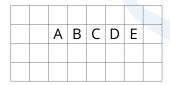
Troo

....

Recap



```
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?



types

Queue:

Stack

Othe

rree

кесар



```
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?
 - Remove element from the array.



Abstract dat types

Queues

Charles

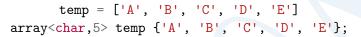
Jets

Othe

Tree

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.
 - Move next element to occupy the empty space.

Abstract dat types

Queue

Ctack

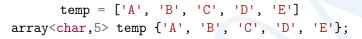
Sets

Othe

Tree

Recap

Coventry University



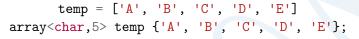


- Array in memory, multiple elements in a contiguous block.
- How do we remove elements from the middle?
 - Remove element from the array.
 - Move next element to occupy the empty space.
 - Repeat.





- 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.





- 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.

0.....

. .

Sets

Othe

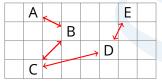
Tree

Quiz

Recap







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



Abstract dat

types

Queue

Stack

5000

Othe

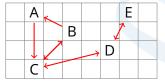
Tree

Quiz

Recap







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



Abstract dat

Ougues

Stack

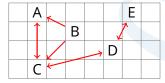
Sets

Othe

Tree

Quiz

Recap



Removing linked list elements

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



Abstract dat

types

Queue:

Stack

Coto

Otho

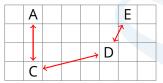
Two

Hee

D ----



Removing linked list elements



- 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 list

Abstract dat

types

Queue

Stack

Sets

Othe

iree

Pocal

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.

Data structures

Data structures

types

Queue:

Stack

Otho

Troo

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

Array example

Abstract data

types

Queues

Stack

Jets

Otne

Tree

Recap

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

Software engineering principal.



Arrays

Array example LL example

Abstract data

types

Queue:

Stack

Sets

Othe

Troo

Quiz

Recap

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



Abstract data types

Abstract data

types

Queue

Stack

rree

Pocan

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



Abstract data types

Abstract data

types

Queues

Stack

Oth

Troo

nee

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.



Abstract data types

Abstract data

types

Queue:

Stack:

Sets

Oth

Tree

Ouiz

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.
- Internals of ADTs can vary widely between implementations.





structs & types

David Croft

122com Data

Array

Array example

Data

Abstract data types

Queue

Stack

Othe

_

. .

Recap

Imagine an ADT like a car.

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





David Croft

Arrays

Linked lists
Array example
LL example

structures
Abstract data

types

Queues

Stack:

Othe

Oui

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.



Arrays

Linked lists
Array example
LL example

structures

Abstract data types

Queues

Stack

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.
- Don't care if, internally, it's using a combustion engine or an electric motor.



Abstract data

types

Queue:

Stack

Sets

Othe

Tree

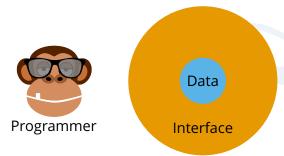
Oui

Recap

Coventry University

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.



Coventry University

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.

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.



Arrays Linked list

Data

Abstract data

Queues

Queue

Stacks

Othe

Tree

Dagar

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?

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.

types

Queues

Stack

...

Quiz

Recap



Array as a queue.

1

front ⇒ _____

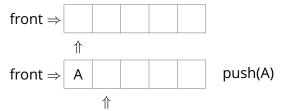
- Very similar to stacks.
 - Keep track of next free space.
 - Limited size.

Stack

Quiz



Array as a queue.



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

Stack

Quiz

Recap

Array as a queue.





 \uparrow

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



Abstract data

Queues

Stacks

<u>.</u>

Othe

Traa

Ouiz

Recap

Array as a queue.



 \uparrow

В

push(C)

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



front \Rightarrow

Otho

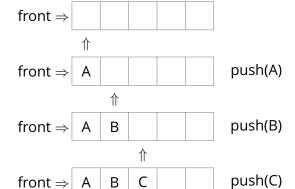
Tree

Ouiz

Recap

Array as a queue.





1

- 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.

Linked lists Array example

Data

Abstract data

Queues

Stacks

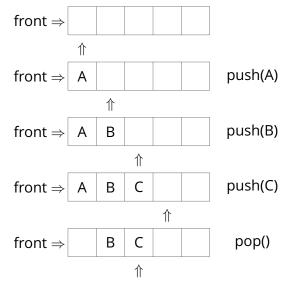
Sets

Othe

Tree

Quiz

пссар



- 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.



Queues



Array as a queue.

- $front \Rightarrow$

front $\Rightarrow \mid A$

1

front \Rightarrow В push(B)

push(C) front \Rightarrow В

push(A)

pop() front \Rightarrow 1

- 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

Stacks

Sets

Othe

Tree

Ouiz

Necap



Array as a queue.

- $front \Rightarrow$
 - 11

front \Rightarrow A

push(A)

 \uparrow

front \Rightarrow A B

push(B)

 \uparrow

front \Rightarrow A B C

push(C)

 \uparrow

front \Rightarrow B C \uparrow

pop()

- 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

structures

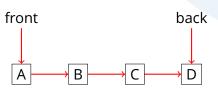
types

Queue

. .

Recap

Linked list as a queue.





Array:

Linked lists

LL example

structures

pop()

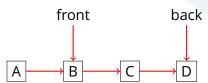
Abstract data types

Queues

Stacks

...

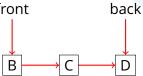
Recap





pop()







structures

Abstract data types

Queues

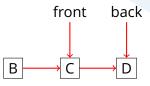
Stacks

Otho

Oui:

Recap

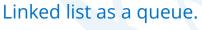
pop() , pop()

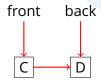




pop(), pop()









Queues

Stack:

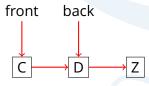
. . .

Otilie

Recap



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





Abstract data types

Queues

Stacks

Otho

A . . : .

Recap





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





David Croft

Stacks

Coventř

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.

Stacks

Abstract dat

types

Queues

Stacks

Juck

Sets

Otile

Tree

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

Coventry University

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

types

Queue

Stacks

sets

Othe

1166

Pocan









- 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.



types

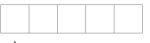
Queue

Stacks

Quiz

кесар









push(A)



- 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 structures

Abstract dat types

Queues

Queue

Stacks

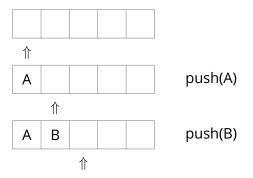
Othic

1166

Recap

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.



Stacks

Α

push(A)

 \uparrow

Α В

push(B)

Α В

push(C)







- 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 Linked list

Array example

structures

Abstract data types

Queues

Stacks

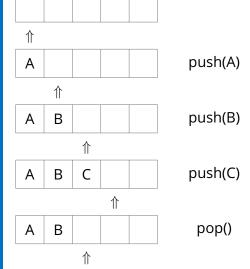
Juck.

Othe

Troo

Ouiz

Recap







- 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.



Structures
Abstract da

Quality

Queue

Stack

Sets

Oth

Tree

Recar



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 C

- 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)

types

Queue

Stack

Sets

Oth

Tree

Recap

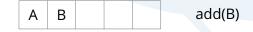


- Items ordered by the set.
- You have no control over it.
- Sets contain unique elements.
 - Can't contain duplicates.
- Can add items to a set.
- Can remove items from a set.
- Can see if an item is in a set.
- Can't get the *n*th element.
 - It's unordered remember.











types

Queue

Stack

Sets

Oth

Tree

Recar

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.





















Coventry University

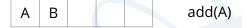
Sets

- 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.











...and the others

Arrays

Array example

structures

Abstract data types

Queue

Stack

Sets

Other

Tree

Recan

Lots of other ADTs.

■ Different names in different languages.

Lists.

Circular lists.

Associative arrays.

■ Dictionaries/Maps.

Double-ended queues.

Trees.

Graphs.



122com Data structs & types

David Croft







Trees

Data structure

Abstract data types

Queue

Stack

Sets

Othe

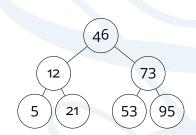
Trees

Quiz

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.





Data structures

Abstract data types

Queue:

Stack

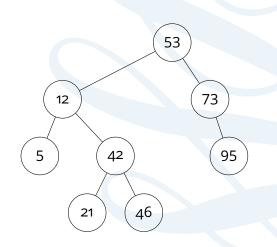
Sate

Othe

Trees

Recap

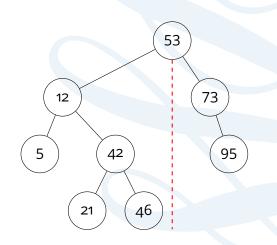
- Trees can be balanced or unbalanced.
- Not required for all trees.
- Going to be talking about BSTs from here on.
- Unbalanced because more than a one node difference between the two halves.





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...





Data structures

Abstract data types

Queue:

Stack

Sets

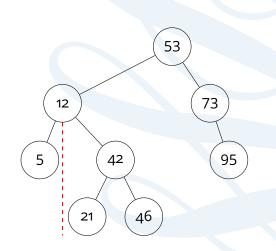
Othe

Trees

O. .:-

Recap

- 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.





Data structures

Abstract data types

Queue

Stack

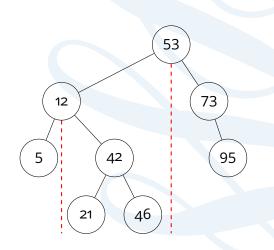
Sets

Othe

Trees

Recap

- 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

Ougues

Jeuc.

0.1

Othic

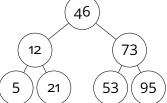
Trees

Recap

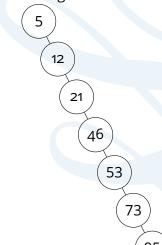
Coventry University

Important that you keep your BSTs balanced.





Degenerate tree.





122com Data structs & types

David Croft

Arrays

Linked lists

Data

Abstract dat

Queues

Stack

Otho

Quiz

Recap



Quiz

David Croft

Arrays

Array example

structures

Abstract dat types

Ougues

Stacks

Jets

Othe

Tree

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

Stacks

5005

Othe

Tree

Quiz

кесар

Stacks and queues are examples of _____

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



Arrays

Linked lists
Array example
LL example

structures

Abstract dat types

Queues

. C+--l--

Stacks

O.1

-...

Quiz

Recap

Coventry University

One advantage of linked lists over arrays is that _____

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

David Cit

Linked list

LL example

structures

Abstract dat types

Queues

. .

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.



Arrays

Array example

Data structure

Abstract data

21

Queue

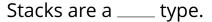
Stack:

Coto

Otho

Troo

Ouiz



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



Arrays

Array example

LL example

structures

Abstract data types

~ ----

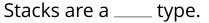
Stack

Sets

Othe

Troo

Quiz



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



Arrays

Array example

Data structur

Abstract dat

OHAHAS

~----

Stack

Sets

Othe

Tree

Quiz

Recap

The elements of a set can't ____

- 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

A....

queue

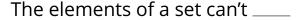
Stack

Jets

Othe

Tree

Quiz



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



Arrays

Linked lists
Array example
LL example

structures

Abstract data types

Ougues

Queue.

Stacks

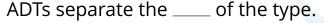
Soto

Othe

Tree

Quiz

кесар



- 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.



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.



Arrays

Linked list Array example LL example

structures Abstract dat

Abstract dat types

Queue

Stacks

Other

Tree

Recap

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.



Array

Array example LL example

structures

Abstract dat

Abstract data types

Queues

Charles

Sets

Othe

Tree

Quiz

Recap

- Complete the yellow Codio exercises for this week.
- Attempt the green Codio exercises for next week.
- If you have spare time attempt the red Codio exercises.
- If you are having issues come to the PSC.

https://gitlab.com/coventry-university/programming-support-lab/wikis/home



122com Data structs & types

David Croft

Arrays

Linked lists

Data

Abstract data types

Queue:

Stacks

Jeach

0.1

Othe

. .

Recap

The End

