Linked lists

LL example

Data structures

Abstract dat types

Oueue:

Charles

Jeach.

Othic

Tree



### 122com Data structures and types

Coventry University



### Overview

Arrays

Linked lists
Array example
LL example

structures

Abstract data types

Queues

Stacks

Sets

Other

Ouiz



- 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

Array example

structures

Abstract data types

Queues

Stacks

Othe

Tree

<u>. . . : -</u>

Recap

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

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

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



#### 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 
$$\leftarrow A \leftarrow B \leftarrow C \rightarrow D$$

$$| \rightarrow |$$
 $\leftarrow$ 

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

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



### Linked lists II



### Arrays

Linked lists Array example

Data structure

Abstract data types

Queues

. .

Stacks

Othe

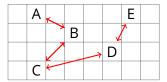
Tree

~ ----

Coventry University

#### 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 II



#### Arrays Linked lists

Array example

Data structures

Abstract data

Queues

. .

Sets

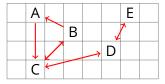
Othe

Tree



### 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 II



#### Arrays Linked lists

Array example

Data structure

Abstract data

Queues

Stacks

Othe

Tree

Q 0.1.2



#### 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;
};
```





Array

Linked lists
Array example

Data structures

Abstract da types

Queues

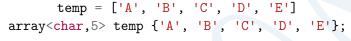
c. .

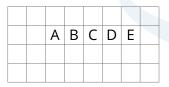
sets

Othe

Tree

Quiz





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





Array

Linked lists

Array example

LL example

Data structures

Abstract da types

Queues

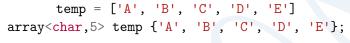
Stack:

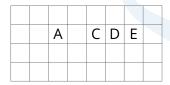
Sets

Othe

Tree

Quiz





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





Array

Linked lists

Array example

LL example

Data structure

Abstract dat types

Queues

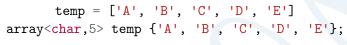
Juack.

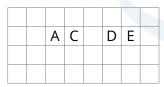
Otne

Tree

D - ---

Coventry





- 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.



Array

Array example

Data structure

Abstract data

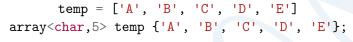
Ouelles

...

Otne

Tree

Ouiz





- 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

Array example

Data structure

Abstract data

Ouelles

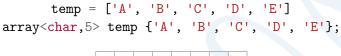
. .

Sets

Othe

Tree

1100





- 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:

Linked lists

Array example

LL example

Data structure

Abstract dat types

Oueue:

Ctack

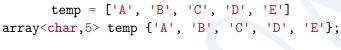
C-4-

Otho

1166

Recar

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.
- Is very slow with large arrays.



Array

Linked list

Array example

LL example

Abstract dat

types

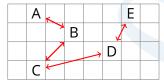
Queue:

Stack

\_ .

Othe

Tree



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





Array

Linked list Array example LL example

Abstract dat

types

Queue:

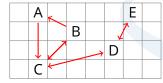
Stack

...

Two o

1100

Pocar



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





Array

Linked list Array example LL example

Abstract dat

types

Queues

Stack

rree



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





Array

Linked list Array example LL example

Abstract dat

types

Queue

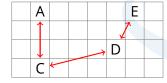
Stack

Sets

Othe

Troc

Ouiz



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



Linked lists
Array example
LL example

Abstract dat

0.....

Queue

Stack

Sets

· ....

1166

Recar



### 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 20<sup>th</sup> 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

types

Oueue:

Stack

Sets

Othe

Tree

~ ----

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

Linked lists Array example

structures

Abstract data types

Queue:

C4--1--

Oth

Ouiz

Recap

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

■ Software engineering principal.





Array

Linked lists
Array example

structures

Abstract data types

Queue:

Stack

Othe

Two

Ouiz

Recap

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





Array

Linked lists
Array example

structures

Abstract data

Abstract dat types

Queue:

Stack

Jiacr

\_\_\_

Tree

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





Array

Array example

structures

Abstract data types

Queues

~----

Stack

. . .

**.**...

....

Recap

- 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.



1

Array:

Array example

structures

Abstract data types

Queues

~----

Stack

Sets

Oth

Tree

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.



# Abstract data

Imagine an ADT like a car.

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



# Abstract data

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.



Linked lists
Array example

structures

Abstract data

types

Queue:

Stack

Sets

Othe

Tree

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

Abstract data

types

Queues

Stack

Sets

Othe

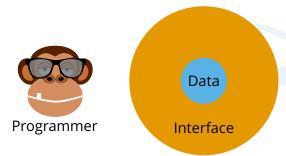
nee

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.



Linked lists
Array example
LL example

Abstract data

Ougues

Queues

Stacks

----

Oth

Tree

Ouis

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.



Linked lists
Array example
LL example

structures
Abstract data

types

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.



Linked lists
Array example
LL example

structures
Abstract dat

types

Queues

Stacks

Sets

Othe

Tree

Docar

#### 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

#### 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 gueue 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.

Data structures

types

Queues

Stack

Sets

Othe

Tree



front  $\Rightarrow$ 

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

Linked lists
Array example

Data structur

Abstract data

Queues

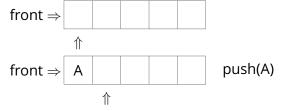
Stack

Sets

Othe

Tree

Recan



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



Linked lists

Data

Abstract data

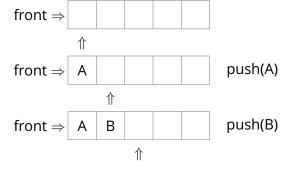
Queues

Stacks

Sate

Othe

Troc



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



#### Array:

Linked lists
Array example

Data structure

Abstract data

Queues

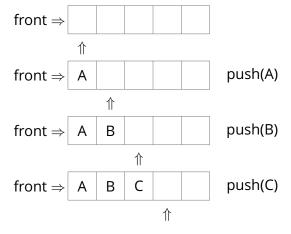
Stacks

Sets

Othe

Troo

Ouiz



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



#### Array

Linked lists
Array example

Data structure

Abstract data

Queues

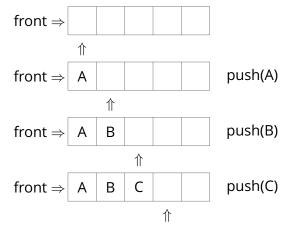
C+--l--

\_\_\_\_

Otho

Troo

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

Linked lists Array example

Data structure

Abstract data

Queues

Stack

Sets

Othe

Tree

Quiz

Coventry University

- $front \Rightarrow$ push(A) front  $\Rightarrow \mid A$ 1 front  $\Rightarrow$ В push(B) push(C) front  $\Rightarrow$ В pop() В 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.

#### Array

Linked lists
Array example

Data structure

Abstract data

Queues

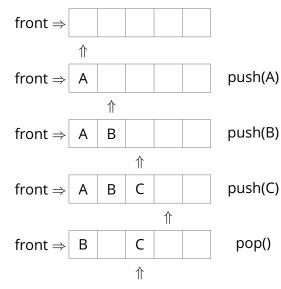
Stacks

C-4-

Othe

Tree

Ouis



- 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

Linked lists
Array example

Data structure

Abstract data types

Queues

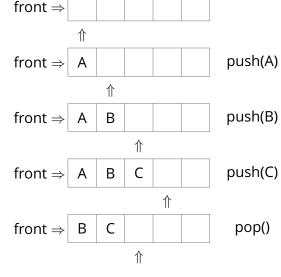
Stacks

Coto

Othe

Tree

Quiz



- 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.





#### Arrays

Linked lists

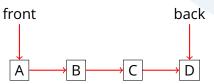
Data

Abstract data

A ....

Queue

Stacks







#### Arrays

Linked lists

LL example

structures

Abstract data

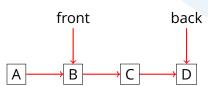
pop()

Queues

. .

Othor

Ouiz







#### Arrays

Linked lists

LL exampl

structures

pop()

Abstract data types

Queues

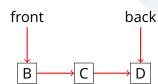
Stacks

\_ .

Other

Troo

Quiz





Linked lists

LL exampl

structures

Abstract data types

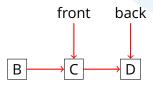
Queues

Stacks

Otho

Recap

pop(), pop()







Array:

Linked lists

LL example

Data

Abstract data

pop(), pop()

-51---

Queue

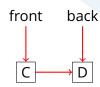
Stack:

Sets

Othe

Tree

Quiz





Linked lists

LL exampl

Abatra et date

Abstract data types

Queues

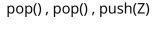
Stack

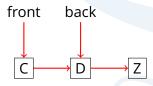
\_\_\_\_

. . .

Otne

11.00







Linked lists

LL example

structures

Abstract data types

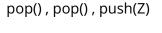
Oueues

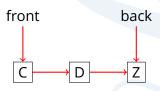
Stack

Juck.

. . .

Ouiz







Linked lists
Array example
LL example

structures
Abstract data

Abstract data types

Oueues

Stacks

Jeden.

Otha

\_

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.



Linked lists
Array example
LL example

structures
Abstract dat

types

Queues

Stacks

Jeack

Sets

Othe

Tree

Dagan

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?



Linked lists
Array example
LL example

structures
Abstract data

types

Queues

Stacks

Stack

Sets

Oth

Tree

Dogge

## 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.

## Array as a stack.

Arrays

Linked lists
Array example

Data

Abstract dat

турсз

Queue

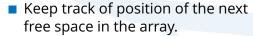
Stacks

Coto

Othe

Tree





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



Linked lists Array example

Data structure

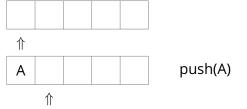
Abstract data

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.



Linked lists Array example

Data

Abstract data

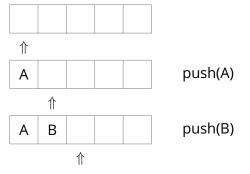
Queues

. . . . . .

Stacks

Othe

Ouiz



- 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.



## Array as a stack.



#### Array

Linked lists Array example

Data structure

Abstract data

Queues

Queue.

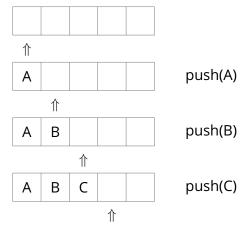
Stacks

5005

Otne

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.



## Array as a stack.

# 1

#### Array

Linked lists Array example

Data structure

Abstract data

Queues

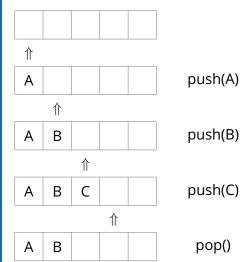
Queue.

Stacks

Otha

Tree

Ouiz



- 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.





Array example

Data structure

Abstract dat types

Queue

Stack

Sets

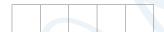
Othe

Tree

Recar



- 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*<sup>th</sup> element.
  - It's unordered remember.







Linked lists
Array example

Data structure

Abstract data

Queue:

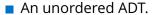
Stack

Sets

Othe

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*<sup>th</sup> element.
  - It's unordered remember.





add(A)





Linked lists
Array example

Data structures

Abstract data types

Queues

Stack

Sets

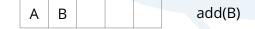
Othe

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.
- Can't get the  $n^{th}$  element.
  - It's unordered remember.











Linked lists
Array example

structures

Abstract dat types

Queues

Stack

Sets

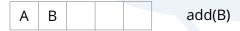
Othe

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.
- Can't get the  $n^{th}$  element.
  - It's unordered remember.













Linked lists
Array example

Data structure

Abstract dat types

Queue:

Stack

Sets

Otne

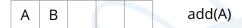
Tree

- 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.
- Can't get the  $n^{th}$  element.
  - It's unordered remember.













Linked lists Array example

#### Data structure

Abstract data

Queue

. .

Stack

.

Other

Tree

Docar

Lots of other ADTs.

■ Different names in different languages.

- Lists.
- Circular lists.
- Associative arrays.
  - Dictionaries/Maps.
- Double-ended queues.
- Trees.
- Graphs.



Linked lists

LL example

structures

Abstract dat types

Queue

Stack:

. . .

Otnei

Trees







Linked lists
Array example

### Data

Abstract data

Ougues

Charles

Jeack.

Otho

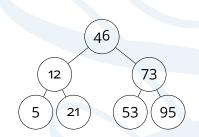
Trees

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.





## Balance



Array

Linked lists Array example

Data structure:

Abstract data types

Queues

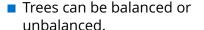
Stack

- .

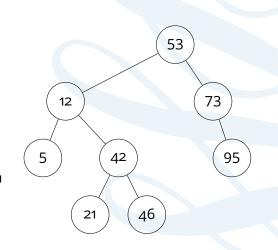
Other

Trees

Q 0...



- 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



#### Array

Linked lists Array example

Data structure

Abstract data

Oueues

Ctools

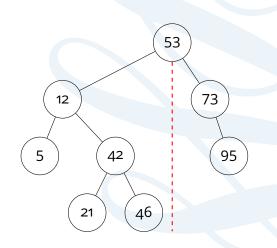
- .

Othe

Trees

Quiz

- 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...







Linked lists Array example

Data structures

Abstract data

Queue

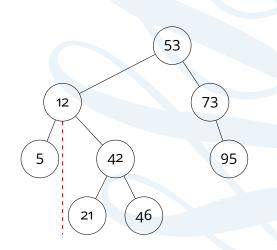
Stack

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...
  - ...and one of the subtrees.





## Balance



#### Array

Linked lists
Array example

Data structure

Abstract data

Oueue

Stack

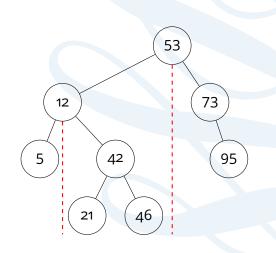
Coto

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.
  - For the whole tree...
  - ...and one of the subtrees.







Linked lists
Array example

Data structures

types

Queues

Stacke

sets

Othe

Trees

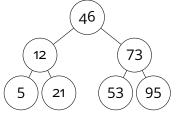
Quiz

Recap

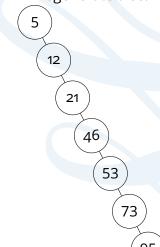


### Important that you keep your BSTs balanced.





### Degenerate tree.





#### Arravs

Linked lists

Array example

Data structures

Abstract dat types

Queue:

Charles

Jeach

....

Othic

O. .:-

Recap



# Quiz

Array example

structures

Abstract data

~ -----

Stacks

Soto

Othe

Tree

Quiz

Recap

## Stacks and queues are examples of \_\_\_\_\_

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



Array example

Data structure

Abstract data

0.....

. . . . . .

Stacks

Sets

Othe

Tree

Quiz

Recap

## Stacks and queues are examples of \_\_\_\_\_

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



Linked lists
Array example
LL example

structures

Abstract dat types

Oueues

\_\_\_\_

Stacks

Othe

Troo

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.

# Linked lists Array example

Data

Abstract dat

Onenes

- .

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.



Linked lists

LL example

structures

Abstract data types

Ougues

Stack

5005

Othe

Tree

Quiz

Recap

## Stacks are a \_\_\_\_ type.

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



Linked lists

Array example

Data structure

Abstract data types

Queuc

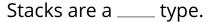
Stack

Sets

Othor

Troo

Quiz



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



Array example

Data structures

Abstract data

Ougues

queue

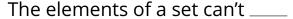
Stack

Cata

Othe

Tree

Quiz



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



Array example

Data structure

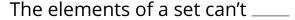
Abstract data

Ougues

Queue

Stack:

Quiz



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



Linked lists
Array example
LL example

Data structures

Abstract data

Ougues

Queues

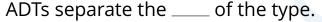
Stacks

C-4-

Othe

Tree

Quiz



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



#### Arrays Linked lists Array example

Data structure

Abstract data

0.,,,,,,,

Queues

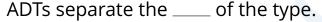
Stacks

Sets

Othe

Tree

Quiz



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



## Why do I care?

#### Arrays

Array example

Data structure

Abstract dat types

Oueues

Queue

Stacl

Sets

Othe

Tree

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.



## Recap

Arrays

Linked list:
Array example

Abstract dat

Abstract dat types

Queue

Stacks

Other

Tree

- 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 example

structures

Abstract data types

Queues

~----

Stack

O. 4. 1- -

. .

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



Linked lists

Data

Abstract data types

Queues

Stacks

Juck

...

Othlei

Quiz



