

A – level Further Mathematics



Many students purchase their own copy of the text books that we use





Edexcel AS and A level Further Mathematics Decision Mathematics 1 Textbook + e-book

Edexcel AS and A level Further Mathematics Core Pure Mathematics Book 1/AS Textbook + e-book

•ISBN: 9781292183299

•ISBN: 9781292207254

A level Further Mathematics – Model 2.1K								
	Core Pure Mathematics:	Decision Mathematics 1:	Decision Mathematics 2:					
Autumn 1	 Complex numbers Series Algebra and functions Calculus Matrices 	 Algorithms and graph theory Algorithms on graphs I 	 Transportation Problems Allocation (Assignment) Problems (AS + AL) 					
Autumn 2	 Matrices Proof Vectors Complex numbers 	 Algorithms on graphs I Algorithms on graphs II (AS + AL) Linear programming 	• Flows in Networks (AS + AL)					
Spring 1	 Further algebra and functions (series) Further calculus Polar coordinates 	• Linear programming	 Dynamic Programming Game Theory (AS + AL) 					
Spring 2	 Polar coordinates Hyperbolic functions Differential equations 	• Critical path analysis	 Game Theory Recurrence Relations (AS + AL) 					
Summer 1	 Complex numbers Series Algebra and functions Calculus Matrices 	• Critical path analysis	• Decision Analysis					
Summar 2		Revision Formal examination	evision					
Summer 2		Formal examination						

Now Some Maths!

Lesson 1

A minimum spanning tree (MST) or minimum weight spanning tree is a subset of the edges of a connected, edge-weighted (un)directed graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight.

Minimum spanning tree - Wikipedia https://en.wikipedia.org/wiki/Minimum_spanning_tree



For these spanning trees from a more complex network Can you come up with the best tree in the sense of minimising the total cost?



Extension: Can you come up with an improved one from the information given? Can you use the trees to build up a possible original network?

Sheet 2

Work in pairs to find a minimum spanning tree for this network – which pair can find the lowest total?

Minimum Connector Algorithms

Kruskal's algorithm

- 1. Select the shortest edge in a network
- 2. Select the next shortest edge which does not create a cycle
- 3. Repeat step 2 until all vertices have been connected

Prim's algorithm

- 1. Select any vertex
- 2. Select the shortest edge connected to that vertex
- 3. Select the shortest edge connected to any vertex already connected
- 4. Repeat step 3 until all vertices have been connected

Example

A cable company want to connect five villages to their network which currently extends to the market town of Avonford. What is the minimum length of cable needed?



We model the situation as a network, then the problem is to find the minimum connector for the network



Kruskal's Algorithm



Kruskal's Algorithm







Select the next shortest











Α

Select the shortest edge connected to that vertex

AB 3



Select the shortest edge connected to any vertex already connected.

AE 4



Select the shortest edge connected to any vertex already connected.

ED 2



Select the shortest edge connected to any vertex already connected.

DC 4



Select the shortest edge connected to any vertex already connected.

CB 5 – forms a cycle

EF 5



Some points to note

•Both algorithms will always give solutions with the same length.

•They will usually select edges in a different order – you must show this in your workings.

•Occasionally they will use different edges – this may happen when you have to choose between edges with the same length. In this case there is more than one minimum connector for the network.



Edges in order of increasing length:

Peel to St Johns 2.7		Douglas to St Johns 8.2
Ballaugh to Kirk Michael Bride to Ramsey 4.6	2.8	Ballaugh to Bride Castletown to St Johns 9.3 Laxey to Ramsey
Castletown to Port Erin	4.7	Castletown to Douglas 10.2
Ballaugh to Ramsey 6.5 Kirk Michael to Peel 6.8		Peel to Port Erin 13.9 Douglas to Ramsey 15.4
Kirk Michael to St Johns	7.4	
Douglas to Laxey 7.7		



	B1	B2	С	D	K	L	P1	P2	R	S
B1	-	9.3	-	-	2.8	-	-	-	6.5	-
B2	9.3	-	-	-	-	-	-	-	4.6	-
С	-	-	-	10.2	-	-	-	4.7	-	9.3
D	-	-	10.2	-	-	7.7	-	-	15.4	8.2
К	2.8	-	-	-	-	-	6.8	-	-	7.4
L	-	-	-	7.7	-	-	-	-	9.3	-
P1	-	-	-	-	6.8	-	-	13.9	-	2.7
P2	-	-	4.7	-	-	-	13.9	-	-	-
R	6.5	4.6	-	15.4	-	9.3	-	-	-	-
S	-	-	9.3	8.2	7.4	-	2.7	-	-	-

Writing the network as an adjacency network:



Starting from any vertex – say B1:

Prim's Algorithm results in the same minimum spanning tree, but the order the edges are added to the tree is different

Ballaugh to Kirk Michael	2.8	
Ballaugh to Ramsey	6.5	
Ramsey to Bride		4.6
Kirk Michael to Peel	6.8	
Peel to St Johns		2.7
St Johns to Douglas	8.2	
Douglas to Laxey		7.7
St Johns to Castletown	9.3	
Castletown to Port Erin	4.7	



Answers to Theme Park Paths

Using Kruskal's

River Rapids to Log Flume Kiddie Corner to Tea Cup Entrance to Pirate Ship Entrance to Animal Farm Tea Cup to Pirate Ship Pirate Ship to Corkscrew Corkscrew to Runaway Train Pirate Ship to River Rapids



The total length of pathway in the minimum connector = 32 + 36 + 48 + 50 + 50 + 58 + 60 + 64 = 398 metres

Extensions

Further work on minimum connector problems can be found at www.mymaths.co.uk website, requires subscription

http://www.mei.org.uk/ MEI website, requires subscription http://www-b2.is.tokushima-u.ac.ip/~ikeda/suuri/diikstra/Prim.shtml (graphical demonstrations of Prim's algorithm) http://www-b2.is.tokushima-u.ac.ip/~ikeda/suuri/kruskal/Kruskal.shtml

(graphical demonstrations of Kruskal's algorithm)

Lesson 2

Chinese postman problems



College Open Day

Chris is delivering leaflets to houses near the college.

The leaflets tell residents about the college open day and apologises for any inconvenience caused.

Chris wants to deliver the leaflets as efficiently as possible.

This is an example of a **Chinese postman** problem.



In this activity you will learn how to solve such problems.

The map shows streets near the college.

Think about

Is it possible to deliver leaflets to the houses in these streets without travelling along the same street twice?



The college map is an example of a **graph**. Here are some others.

A graph is made up of a collection of vertices called **nodes**, joined by edges called **arcs**.

A **traversable** graph is one that can be drawn without taking a pen from the paper and without retracing the same edge







Think about

Which of the graphs above can be traced without taking your pen from the paper?

A Swiss mathematician, Leonhard Euler (1707 to 1783), published the first paper on graph theory in 1736.

The paper was based on the Seven Bridges of Konigsberg problem.

Residents of Konigsberg wanted to know if it was possible to find a route which passed across each of the bridges only once.



Euler found that the **order of the vertices** determines whether or not a graph is traversable.

If it is possible to traverse a graph starting and finishing at the same point then the graph has an **Eulerian trail**.

If it is possible to traverse a graph starting at one point and finishing at a different point then the graph has a **semi-Eulerian trail**.





- A order 4
- B order 4
- C order 5
- D order 2
- E order 4
- F order 4
- G order 3

The only odd vertices are C and G



An Eulerian trail is only possible if all vertices are even.

Think aboutCan you explain why?

A semi-Eulerian trail is possible:

This started at C but ended at G.

The postman needs to return from G to C by the **shortest** route.

The **shortest** route is GE + EC.

The total distance travelled to deliver the leaflets

= 1691 m



College

100m

60m

110m

130n

80m

82m

G

175m

80m

72m

The Chinese postman algorithm

Step 1 Find all the **odd** vertices in the network.

Step 2 Consider all the routes joining **pairs of odd vertices**. Choose the routes with the **shortest total distance**.

Step 3Add in these edges again.This will give a network with only even vertices.

Step 4 Find an Eulerian trail.

Try this – Easter Parade

Easter Parade

- A order 3
- B order 4
- C order 3
- D order 2
- E order 5
- F order 4
- G order 3
- H order 2



Odd vertices: A, C, E, G Possible pairings: AC + EG = 9 + 8 = 17 mins AG + EC = 13 + 7 = 20 mins

AE + CG = 5

+ 11 = **16 mins**



The smallest total is: AE + CG = 16 min

Add these edges to the network

The total time in the original network = **97 min**



Shortest time = 97 + 16 = 113 minutes = 1 hour 53 minutes

Possible Eulerian trail: ABDFBEAEFHGECGCA

Chinese postman problem

Reflect on your work

An Eulerian trail is a path which starts and ends at the same vertex and includes every edge just once. Euler discovered that such a trail will exist if and only if all vertices are even.

Can you explain why?

In any network, the sum of the orders of the vertices is even. Can you explain why?

In any network, the number of odd vertices must be even. Can you explain why?

Points for discussion

Why all the vertices have to be even in order for the graph to be Eulerian.

Why the sum of the vertices on any network will be even.

Why, in any network, the number of odd vertices must be even.

You may also like to include that:

• when finding a route, the number of times each vertex is visited is (order \div 2), except for the start/finish which is visited (order \div 2) + 1.

• the number of ways of pairing n odd vertices is $1 \times 3 \times 5 \times 7 \times ... \times (n-1)$

• as the number of odd nodes increases, the number of possible pairings to be considered increases dramatically. It is therefore impractical to do the algorithm by hand with more than 6 nodes. Even with a computer, it can become too time consuming – with 12 odd nodes there are 10395 combinations to consider.

Extensions

Students can be asked to check whether their Eulerian trail conforms to the rule for the number of times each vertex should occur (given above).

Use a Search engine to find other Chinese postman problems on the internet.

Holiday Task

- Over the summer we would like you to research complex numbers and create a poster about them.
- Your poster can be on A4 or A3 and could feature the mathematics, history and applications of complex numbers.
- The poster should be brought to your first Further Maths lesson.
- You should spend at least 5 hours researching and creating your poster

This activity shows you how to use Kruskal's and Prim's algorithms to solve minimum connector problems. This happens when you want to connect a network and minimise the cost.

Information sheet

The map shows some of the towns and roads on the Isle of Man.

A cable TV company wants to lay cables to connect the towns, laying the cable along the roads shown on the map.

They want to connect all of these towns to their cable network using the **minimum** total length of cable.

The length of the roads joining adjacent towns is given in miles in the chart below.

Note that a dash in the table (–) means there is no direct route between the towns.



Ballaugh		_							
9.3	Bride		_						
-	-	Castletown		_					
-	-	10.2	Douglas		_				
2.8	-	-	-	Kirk Michael		_			
-	-	-	7.7	_	Laxey		_		
-	-	-	-	6.8	-	Peel			
-	-	4.7	-	-	-	13.9	Port Erin		
6.5	4.6	-	15.4	-	9.3	-	-	Ramsey	
-	-	9.3	8.2	7.4	-	2.7	-	-	St. Johns

To solve the problem you need to find a **spanning tree** of **minimum** length.

A spanning tree is a tree that connects all the vertices together.

A minimum spanning tree is a spanning tree of minimum length

Think about...

What is the connection between the number of towns and the number of edges in the minimum spanning tree?

The type of problem involved in this activity is often called a **minimum connector problem**. **Kruskal's algorithm** is one method of solving such problems.

Kruskal's Algorithm

This is for finding a minimum connector (that is minimum spanning tree).

Step 1 List the edges in order of increasing weights.

Step 2 Start with the edge with the smallest weight.

Step 3 From the remaining edges, choose the one with the smallest weight which does not form a cycle. (If there are 2 shortest edges, choose either.)

Step 4 Repeat Step 3 until all the vertices are connected.

Try this: Cable TV problem

Carry this out for the Cable TV problem. First list the edges in order of length below.

Then follow steps 2 to 4, using the map below to show your solution.

Write down the order you introduce the edges into the minimum spanning tree.





This gives an alternative method for solving minimum connector problems.

Prim's Algorithm for finding a minimum connector (i.e. minimum spanning tree):

Step 1 Starting from any vertex, join it to the nearest adjacent vertex.

Step 2 Join the next nearest vertex to those already included, provided that this does not form a cycle. (If there are 2 nearest vertices, choose either.)

Step 3 Repeat Step 2 until all the vertices are included.

Try this ... using the adjacency matrix (table) below.

First choose a starting vertex, say B_1 , and label it as 1 above the table. Delete row B_1 and look for the smallest entry in column B_1 . This is the 2.8 in row K.

This means B_1K is in the solution. Label K as 2 (in table) and draw B_1K (on map).

Now delete row K and look for the smallest entry in column B_1 or K. This is 6.5 in column B_1 and row R. Label R as 3 and draw B_1R on the map. Continue in this way until all the vertices are connected.

	B ₁	B ₂	С	D	К	L	P ₁	P ₂	R	S
B ₁	-	9.3	1	-	2.8	I	1	-	6.5	_
B ₂	9.3	-	-	-	-	Ι	-	I	4.6	_
С	-	1	1	10.2	I	I	1	4.7	-	9.3
D	I	I	10.2	I	I	7.7	1	I	15.4	8.2
К	2.8	1	1	-	I	I	6.8	-	-	7.4
L	-	1	1	7.7	I	I	1	-	9.3	_
P_1	I	I	1	I	6.8	I	1	13.9	I	2.7
P ₂	I	I	4.7	I	I	I	13.9	I	I	-
R	6.5	4.6	-	15.4	-	9.3	-	I	I	_
S	_	_	9.3	8.2	7.4	_	2.7	_	_	_

Port



You should find the solution is identical to that found using Kruskal's algorithm.

Think about...

Can you explain how the matrix method works?

Which do you prefer – Kruskal's or Prim's algorithm? Why?

What else would need to be considered in the real situation?

Try this: Theme Park Problem

The plan below shows the lengths of paths linking parts of a theme park.



The manager of the theme park wants to widen and re-surface some of these paths to provide better access to the rides for wheelchairs. His aim is to provide better paths to connect all parts of the theme park, but using the minimum total length possible.

Use Kruskal's and Prim's algorithms to find the minimum spanning tree for this network.

Write a paragraph explaining the advantages and disadvantages that your solution may have in the real situation.

Reflect on your work

How many edges will there be in the minimum spanning tree of a network with n vertices?

What are the main advantages and disadvantages of Kruskal's and Prim's algorithms?

What other sorts of things might need to be considered in a real situation?



In this activity you will use the **Chinese postman algorithm**, (also called the Route Inspection Problem), to solve practical problems. It is called the Chinese postman algorithm because it was studied in 1962 by a Chinese mathematician called **Kwan Mei-Ko**, who was interested in minimising the total distance walked by a postman delivering mail. It is based on **Euler's** findings for **traversable** graphs.

Information

A graph is made up of a collection of vertices called **nodes**, joined by edges called **arcs**. A **traversable** graph is one that can be drawn without taking a pen from the paper and without retracing the same edge.

Try this ...

College Open Day

A college is planning an Open Day.

The college car park is small, and visitors are likely to park on streets near the college. The map shows those streets likely to be affected.

The Principal wants to deliver a leaflet to the houses on these streets to tell residents about the open day, and to apologise for any inconvenience caused.

Your task is to find the most efficient way for one person to deliver these leaflets.



Think about...

Do you think it is possible to deliver leaflets to the houses on these streets without travelling along the same street twice? Try out some routes on the map. Start from the college.

The solution of this type of problem depends on whether the vertices in the network are **odd** or **even**. A vertex is odd or even depending on the number of edges that meet there. The number of edges that meet at the vertex is called the **order** or **degree** of the vertex.

To answer

The vertices in the network are listed below.
 For each vertex, state the order and whether it is even or odd.



Think about...

A graph is traversable if all its vertices are **even**, or it has just two **odd** vertices.

If a graph has all even vertices, it is possible to traverse the graph starting and finishing at the same vertex.

Can you explain why?

In such a case, the graph is said to have an **Eulerian trail**. An Eulerian trail is a path which starts and ends at the same vertex and includes every edge just once.

If a graph has two odd vertices, it is possible to traverse it by starting at one of the odd vertices and finishing at the other. In this case the graph is said to be **semi-Eulerian**.

The network of streets near the college has two odd vertices, C and G. It is possible to find a **semi-Eulerian trail** that starts at C, passes along each edge just once, and ends at G.

Think about...

Can you find a path that that starts at C, passes along each edge just once, and ends at G?

To deliver the leaflets to all the streets and return to the college, some streets will be travelled along more than once when the 'postman' returns from G to C.

The most efficient route depends on the lengths of the streets.



Solving the Chinese postman problem

Here is the general method for solving the Chinese postman problem.

- a Identify all the odd vertices in the network.
- **b** List all the possible ways to pair the odd vertices.
- **c** For each pair of odd vertices, find the edges with the minimum weight that connect the vertices.
- **d** Find the pairings for which the total weight is as small as possible.
- e Add these edges onto the original network. You will now have a network with only even vertices.
- **f** Add up the weights of all the edges in the original network and add to this the total weight of the edges you have added. This is the minimum weight of an Eulerian trail.
- g Find a route for the new network there are usually many possibilities.

Try this ... Easter Parade

An Easter Parade is to travel along each street in the network shown below.

The estimated time it will take to travel along each street is shown on the network.

1 For each vertex in the network, state the order and whether it is even or odd.

A..... B..... C..... D..... E.... F.... G..... H....



2 List the vertices that are odd

3 Pair the odd vertices to give the smallest total time

Add in these edges again on the diagram.

4 Write down a route for the Easter Parade that starts and ends at A and goes along each edge once.

5 Calculate an estimate for the total time this will take.

.....

Think about...

What else might you consider in the real situation?

Reflect on your work

An Eulerian trail is a path which starts and ends at the same vertex and includes every edge just once.

Euler discovered that such a trail will exist if and only if **all vertices are even.** Can you explain why?

In any network, the sum of the orders of the vertices is even. Can you explain why?

In any network, the number of odd vertices must be even. Can you explain why?