SUMMARY SHEET – DECISION MATHS

CRITICAL PATH ANALYSIS

The main ideas are covered in

AQA	Edexcel	MEI	OCR
D2	D1	D1	D2

The main ideas in this topic are

- Drawing Activity or Precedence Networks
- Performing Forward and Backward Passes and Identifying Critical Activities
- Drawing Cascade Charts and Resource Levelling

Before the exam you should know

- How to draw precedence networks. as you possibly can.
- When you need to use dummy activities.
- How to perform forward and backward passes on a precedence network to calculate early and late start times.
- How to find the critical activities.
- How to calculate the various types of float.
- How to draw a cascade chart and construct a resource histogram.
- Where resource levelling is required and how to make effective use of float to improve efficiency.
- What is meant by crashing a network.

Terminology

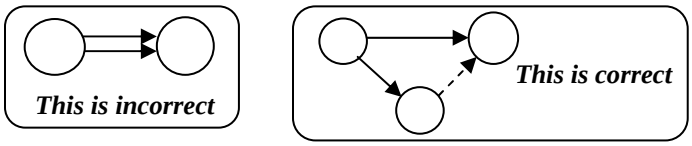
An activity is a task which needs to be done and takes an amount of time/resources to complete.

Precedence tables show the activities that need to be done together with their duration and their immediate predecessors.

Precedence networks show the sequence of the activities**.** The network must have one start node and one end node.

An **event** is the start/finish of one or more activities.

Dummy activities are used to keep the correct logic and to ensure each activity is **uniquely** defined by (i, j) where i is its starting event and j is the finishing event.



It can be a good idea to do an initial sketch as it's often possible to make your diagram clearer by repositioning activities to avoid them crossing over one another**.**

Forward pass establishes the earliest times that events can happen.

Backward pass establishes the latest time that an event can happen.

Critical activites are those whose timing is critical if the project is to be completed in the minimum time. The critical activities will form a path through the network

Float is the amount of time by which an activity can be delayed or extended.

Independent float does not affect other activities.

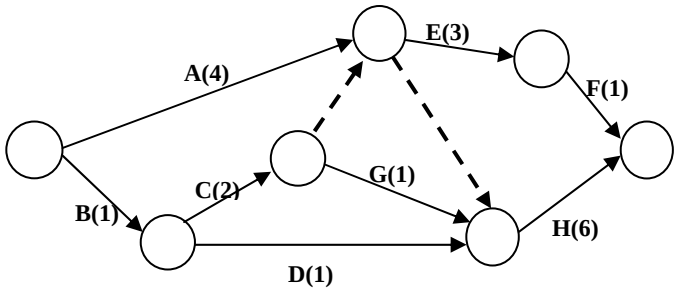
Interfering float is shared between two or more activities.

Example:

The table shows the activities involved in creating a small patio in a garden.

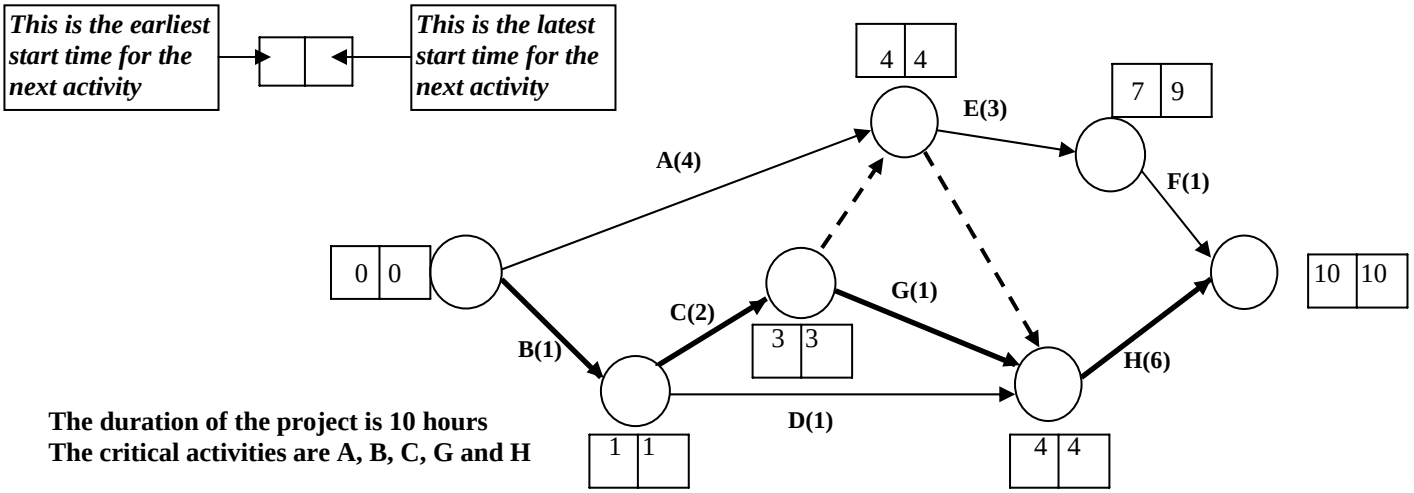
Activity Name	Task	Time (hrs)	Preceding Activities
A	Clear Garden	4	
B	Measure area	1	
C	Design Patio	2	B
D	Purchase fencing	1	B
E	Buy pots and plants	3	A,C
F	Plant all pots	1	E
G	Purchase paving	1	C
H	Construct Garden	6	A, D,G

The network for this precedence table



Disclaimer: Every effort has gone into ensuring the accuracy of this document. However, the FM Network can accept no responsibility for its content matching each specification exactly.

The forward and backward pass



The duration of the project is 10 hours
The critical activities are A, B, C, G and H

Float

activity	float	type
D	2 hours	independent
E	2 hours	Interfering (with F)
F	2 hour	Interfering (with E)

In this example there are two hours of float shared between activities E and F

Cascade Chart and Resources levelling

A Cascade Chart shows each activity set against a time line.
Float time is sometimes shown by using shading.
Dependencies are shown by vertical lines.
The cascade chart can be adjusted by using the float times to make use of resources more efficient.

If activity A needs two people and all the rest can be done by one person, then the resource histogram looks like this (note that 4 people are needed in the second hour).

If only three people are available for the first three hours, but a fourth friend can then come and help for an hour, we could move activity D within its float time to make this possible.

This would make the cascade chart look like this

The resource histogram would now look like this

Disclaimer: Every effort has gone into ensuring the accuracy of this document. However, the FM Network can accept no responsibility for its content matching each specification exactly.

REVISION SHEET – DECISION MATHS

DYNAMIC PROGRAMMING

The main ideas are covered in

AQA	Edexcel	MEI	OCR
D2	D2		D2

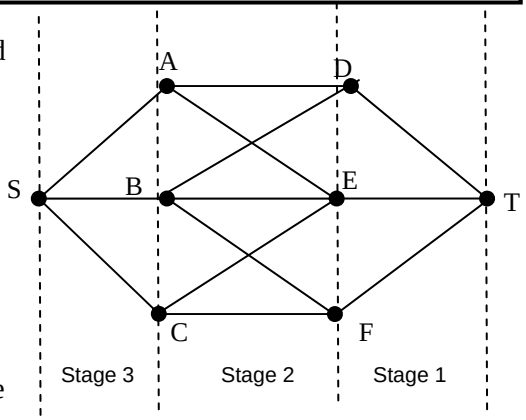
The main idea in this topic is:

- Finding the shortest (or longest) route through a network by working backwards from T to S

Before the exam you should know:

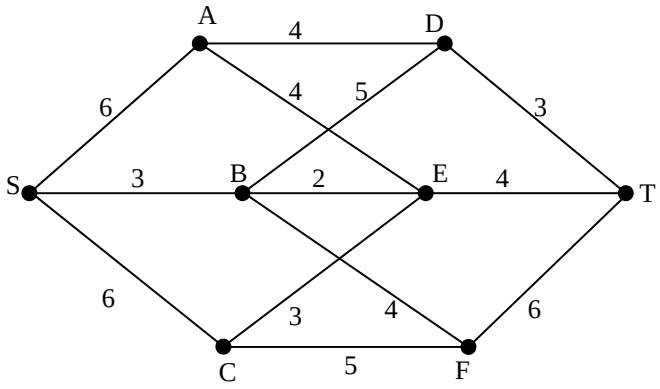
- That dynamic programming is a technique for solving multi-stage decision making problems by working backwards.
- What is meant by stage and state variables, actions and values (or costs).
- The meaning of sub-optimisation.
- How to set up a dynamic programming tableau.
- How to calculate the sub-optimal strategy at each stage.
- That a minimax route is one on which the maximum route is as small as possible.
- That a maximin route is one on which the minimum route is as large as possible.

Dynamic programming is used to solve some optimisation problems modelled By networks, though the solution is usually presented in a table. In a dynamic programming network, the nodes are referred to as **states**, directed arcs are called **actions** and the transition from one state to the next is a **stage**.



The method starts at T (state 0) and considers all the nodes joined directly to T (D, E and F in the diagram), which are called stage 1 nodes.
We find the best route from each of these to T then move on to the stage 2 nodes (in this case A, B and C). The optimal route from each of these to T is found using the best routes from stage 1 and so on until we reach S.
At each stage you work out the best strategy from that point. This is called the **sub-optimal strategy**.

Example 1: The network below shows the weight limit, in tonnes, on vehicles that use the road. A transport company making regular deliveries from S to T wants to use the largest possible vehicle in order to minimise their outgoings. What is the heaviest vehicle they can use?



Optimal route S C F T
Maximum weight of lorry is 5tonnes

The problem is about finding the route with the greatest minimum weight from S to T so it is an example of a maximin problem. For each node we are choosing the minimum arc on the current route to that node.

Solution

stage	State (node)	Action (into node)	Value (Route min)	Current max
1	0 (D)	0	3*	3
	1 (E)	0	4*	4
	2 (F)	0	6*	6
2	0 (A)	0	min (4,3)=3	4
		1	min (4,4)=4*	
	1 (B)	0	min (5,3)=3	4
		1	min (2,4)=2	
		2	min (4,6)=4*	
	2 (C)	0	min (3,4)=3	5
		1	min (5,6)=5*	
3	0 (S)	0	min (6,4)=4	5
		1	min (3,4)=3	
		2	min (6,5)=5*	

Disclaimer: Every effort has gone into ensuring the accuracy of this document. However, the FM Network can accept no responsibility for its content matching each specification exactly.

Example 2

A company is planning to build three new houses A, B and C at the rate of one every three months. They can build the houses in any order but the profit will be affected by the workers available and the supply costs, which are subject to variation. The expected profits in thousands of pounds are shown in the table

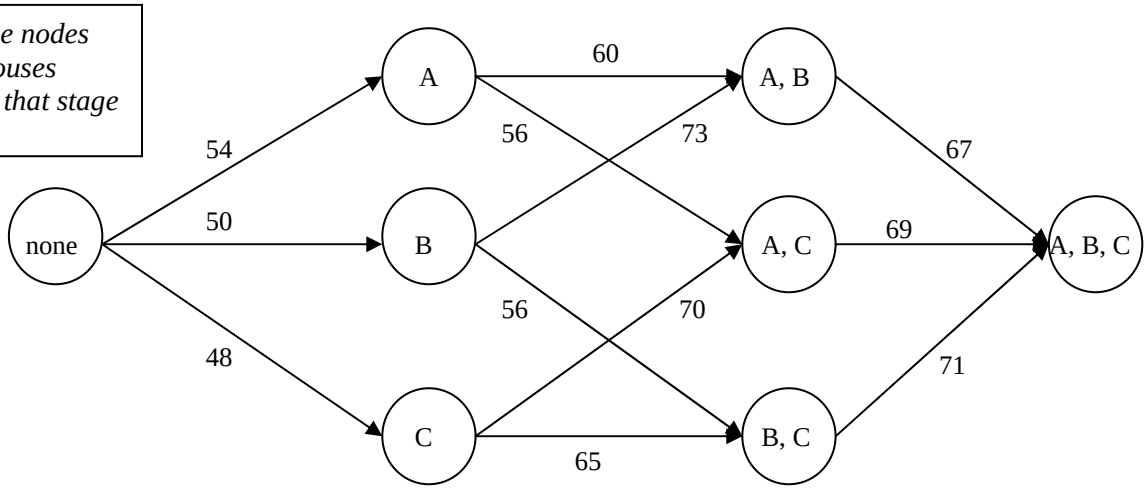
Time period	Already completed	Profits (£1000)		
		A	B	C
1	--	54	50	48
2	A	-	60	56
	B	73	-	56
	C	70	65	-
3	A and B	-	-	66
	A and C	-	69	-
	B and C	71	-	-

(a) Represent the information on a network such that the optimal strategy will correspond to the longest path through the network.

(b) Use dynamic programming to determine the order in which they should build the houses in order to maximise profit

Solution

The letters in the nodes represent the houses already built at that stage of the project



(b)

Stage (time period)	State (houses built)	Action (house to be built)	Value (profit)	max
3	0 (A, B)	0 (C)	67*	66
	1 (A, C)	0 (B)	69*	69
	2 (B, C)	0 (A)	71*	71
2	0 (A)	0 (B)	60+67=127*	127
		1 (C)	56+69125	
	1 (B)	0 (A)	73+67=140*	140
		1 (C)	56+71=128	
	2 (C)	0 (A)	70+69=139*	139
		1 (B)	65+71=136	
1	0	0 (A)	54+127=181	190
		0 (B)	50+140=190*	
		0 (C)	48+139=187	

It can be very helpful to write what is happening in the state and action columns.
In this case the state is the houses that have already been built and the action is the house that is about to be built.

The houses should be built in the order B then A then C

The maximum profit is £190000

Disclaimer: Every effort has gone into ensuring the accuracy of this document. However, the FM Network can accept no responsibility for its content matching each specification exactly.

REVISION SHEET – DECISION MATHS

GAME THEORY

The main ideas are covered in			
AQA	Edexcel	MEI	OCR
D2	D2		D2

The main ideas in this topic are:

- Finding out whether a two person zero sum game has a stable solution.
- If a stable solution does not exist, using a graphical method to find an optimal mixed strategy.

In a two person, **zero sum game**, one player's gain is the same as the other player's loss.

Before the exam you should know:

- What is meant by a two person, zero sum game.
- How to interpret a payoff matrix.
- How to find a playsafe strategy.
- What is meant by a stable solution.
- How to find the value of a game.
- How to simplify games using dominance.
- How to find an optimal mixed strategy for a game with no stable solution.
- How to convert a two person zero sum game into a linear programming problem.

A **pay-off matrix** represents the gain for one of the players for each combination of strategies for the two players in a two-person game.

Finding a play-safe strategy for a zero-sum game

In the pay-off matrix for A, find the minimum entry in each row; A will use the strategy that involves the greatnesses of these values (maximin).Then find the maximum entry in each column, B will use the strategy that involves the lowest of these values (minimax).

A **stable solution** occurs if and only if the maximum(row minimum) = the minimum(column maximum).
A assumes B uses his play-safe strategy and cannot do better by using an alternative strategy and B assumes A uses his play-safe strategy and cannot do better by using an alternative strategy. A stable solution is sometimes called a **saddle point**.

Example: Anna and Barry play a zero sum game. The game can be represented by the following payoff matrix for Anna

	B_1	B_2	B_3
A_1	3	2	5
A_2	-1	4	-2
A_3	2	1	3

(a) Show that there is not stable solution

Solution

	B_1	B_2	B_3	Row max	minimax
A_1	3	2	5	**5**	**3**
A_2	-1	4	-2	**4**	
A_3	2	1	3	**3**	
Col min	**-1**	**1**	**-2**		
maximin	**1**				

$1 \neq 3$ so solution is not stable

Dominance: A row/column can be eliminated if all the entries in that row/column are less than or equal to the corresponding entries of another row/column because the player would never choose that row/column.

Finding a mixed strategy using a graph: If player A has two possible strategies, assume he adopts the first strategy with probability p and the second with probability $1-p$. The expected payoff for A will depend on which strategy B chooses so plot A's expected payoff for each of B's strategies as lines on a graph. A's best strategy is to play with the value of p which gives the highest point at the intersection of the lines. This will maximise A's minimum return (maximin). The value of the game can be calculated using this value of p.

	B_1	B_2	B_3
A_1	3	2	5
A_2	-1	4	-2
A_3	2	1	3

Example (cont)

(b) Explain why Anna should never play strategy A_3

(c) Find the best mixed strategy for Anna and give the value of the game for her.

Disclaimer: Every effort has gone into ensuring the accuracy of this document. However, the FM Network can accept no responsibility for its content matching each specification exactly.

Solution

(b) Anna should not play A_3 since row A_3 is dominated by row A_1 since $3 > 2$, $2 > 1$ and $5 > 3$.

(c)

	B_1	B_2	B_3
A_1	3	2	5
A_2	-1	4	-2

Let Anna play
A_1 with probability p
A_2 with probability $(1-p)$

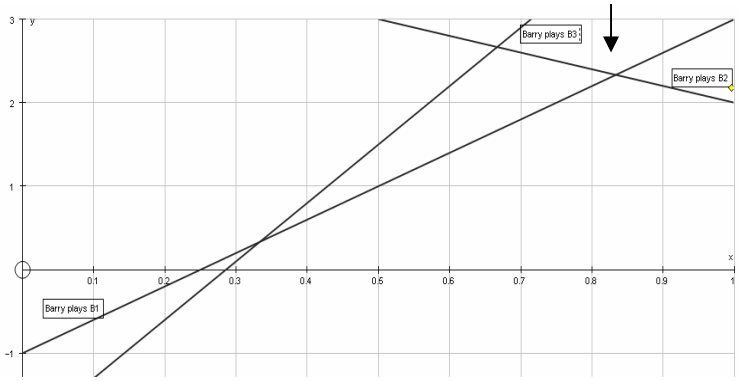
Anna's expected payoff if Barry plays strategy:

B_1: $3p-1(1-p)=4p-1$
B_2: $2p+4(1-p)=4-2p$
B_3: $5p-2(1-p)=7p-2$

Solving for p :

$4p-1=4-2p$
$\Leftrightarrow 6p=5$
$\Leftrightarrow p=\frac{5}{6}$

Anna's optimal strategy



From the graph you can see that the optimal expected return for Anna occurs at the intersection of the lines for Barry playing strategies B_1 and B_2.

So Anna should play A_1 with probability $\frac{5}{6}$ and A_2 with probability $\frac{1}{6}$

Value of the game: $4\left(\frac{5}{6}\right)-1=\frac{20}{6}-1=\frac{14}{6}=2\frac{1}{3}$

Finding a mixed strategy using the Simplex algorithm

The graphical method cannot be used if player A has more than two strategies so we must use the simplex method.

Let the probabilities of each of A's strategies be p_1, p_2, p_3 etc.

If there are any negative elements in the payoff matrix, add k to make them all positive.

Model the situation as a linear programming problem:

Maximise V-k (this is the expected payoff from the original game)
Subject to $V \leq$ expected payoff from B playing strategy 1
$V \leq$ expected payoff from B playing strategy 2 e*tc*
$p_1 + p_2 + p_3 + ... \leq 1$ (because the sum of the probabilities cannot exceed 1)
$p_1,\ p_2,\ p_3 \geq 0$ (no negative probabilities)

This information can now be put into an initial simplex tableau and solved in the usual way.

Disclaimer: Every effort has gone into ensuring the accuracy of this document. However, the FM Network can accept no responsibility for its content matching each specification exactly.

REVISION SHEET – DECISION MATHS

ALLOCATION PROBLEMS AND THE HUNGARIAN ALGORITHM

The main ideas are covered in

AQA	Edexcel	MEI	OCR
D2	D2		D2

The main ideas in this topic are:

- Extending the ideas introduced in matchings to find a maximum matching and its associated cost.
- Using the Hungarian algorithm to find an optimal, minimum cost solution.

Before the exam you should know:

- That you will be working on a matrix or array of numbers called the payoff matrix.
- How to reduce a matrix to one containing zeros and interpret this to give an optimal allocation.
- How to apply the Hungarian algorithm to minimise costs.
- That the matrix of numbers must be square before you can apply the Hungarian algorithm.
- How to deal with maximisation problems by subtracting all the entries in the matrix from a constant.

Example

XYZ taxis have four drivers, Alan, Betty, Chris and Dave. One morning they have four bookings to collect people (P, Q, R, S) and take them to the station for the 0800 train. The table shows the time in minutes it will take for each driver to drive form their house to the customer's house and then transport each customer to the station.

	P	Q	R	S
A	20	20	28	27
B	23	20	32	24
C	27	26	17	23
D	25	24	25	26

The manager of XYZ taxis wants to minimise the total time taken, who should he allocate to each journey?

Allocation problems are solved by reducing the payoff matrix by subtracting the least entry from all the entries in that row (or column). You now have a matrix showing *relative* costs. Each row (column) will now be zero. These zeros will represent a minimum "cost" solution. To have a maximum matching is to find the minimum number of lines needed to cover all the zeros.

Solution: Firstly reduce the rows

	P	Q	R	S	
A	0	0	8	7	Subtract 20 from each entry
B	3	0	12	4	Subtract 20 from each entry
C	10	9	0	6	Subtract 17 from each entry
D	1	0	1	2	Subtract 24 from each entry

This is not a maximum matching since all the zeros can be covered with 3 lines.

	P	Q	R	S
A	0	0	8	7
B	3	0	12	4
C	10	9	0	6
D	1	0	1	2

Reduce columns

	P	Q	R	S
A	0	0	8	5
B	3	0	12	2
C	10	9	0	4
D	1	0	1	0

This is now a maximum matching.
Alan – P, Betty – Q
Chris – R, Dave – S
Total time = 20 + 20 + 17 + 26
= 83 minutes

The Hungarian algorithm

1. If the payoff matrix is not square, add in dummy row(s) or column(s) of equal numbers to make it square.
2. Subtract the minimum entry from each row from all the entries in the row.
3. If necessary repeat step 2 for the columns.
4. Draw the minimum number of horizontal and/or vertical lines to cover all the zeroes.
5. If the number of lines is equal to the number of columns in the matrix, the positions of the zeroes indicate an optimal matching; if not, go to step 6.
6. Augment the matrix: find the smallest element not covered; subtract it from the non-covered elements and add it to any elements covered by 2 lines.

Go to step 2.

Disclaimer: Every effort has gone into ensuring the accuracy of this document. However, the FM Network can accept no responsibility for its content matching each specification exactly.

Note: If you need to maximise, subtract every number in the original payoff matrix from the largest number in the matrix before applying the algorithm. The final payoff is then found by adding the original payoffs in the cells used in the optimal matching.

Example: The average scores for five members of a quiz team are shown in the table:

	music	sport	geography	history	General knowledge
Alan	17	19	18	15	16
Brenda	20	18	15	19	17
Cally	13	17	17	16	14
David	12	16	18	15	14
Edwin	14	16	15	16	15

A different person must be chosen to answer questions in each of the five rounds in the final. Using their past performance, who should do each round in order to maximise score?

(a) Explain why you should replace each entry x by $20 - x$ before using the Hungarian Algorithm
(b) Form a new table by subtracting each number from 20. Use the Hungarian algorithm to allocate the sports team members.
(c) State the expected score for the team based on their practice scores.

Solution: (a) Because the problem is a maximisation problem and 20 is the largest number in the matrix.

(b)

	music	sport	geography	history	General knowledge
Alan	3	1	2	5	4
Brenda	0	2	5	1	3
Cally	7	3	3	4	6
David	8	4	2	5	6
Edwin	6	4	5	4	5

Reduce rows

	music	sport	geography	history	General knowledge
Alan	2	0	1	4	3
Brenda	0	2	5	1	3
Cally	4	0	0	1	3
David	6	2	0	3	4
Edwin	2	0	1	0	1

Reduce columns

	music	sport	geography	history	General knowledge
Alan	2	0	1	4	2
Brenda	0	2	5	1	2
Cally	4	0	0	1	2
David	6	2	0	3	3
Edwin	2	0	1	0	0

Solution is not optimal as all zeros can be covered with 4 lines
Augment by 1

	music	sport	geography	history	General knowledge
Alan	2	0	1	3	1
Brenda	0	2	5	0	1
Cally	4	0	0	0	1
David	6	2	0	2	2
Edwin	3	2	2	0	0

It now takes 5 lines to cover the zeros so matching is optimal

Alan – sport, Brenda – music, Cally – History, David – Geography, Edwin – General Knowledge

(c) Expected scores: 19 + 20 + 16 + 18 + 15 = 88

Disclaimer: Every effort has gone into ensuring the accuracy of this document. However, the FM Network can accept no responsibility for its content matching each specification exactly.

REVISION SHEET – DECISION MATHS

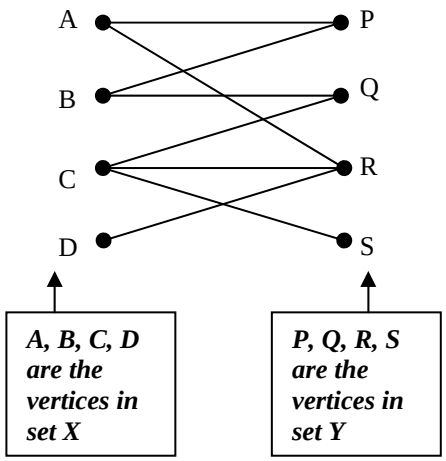
MATCHINGS

The main ideas are covered in			
AQA	Edexcel	MEI	OCR
D1	D1		D2

The main ideas in this topic are:

- Modelling real situations using bipartite graphs.
- Using the maximum matching algorithm to solve problems

A **bipartite graph** has two sets of vertices, X and Y such that the edges only connect vertices in set X to those in set Y and never to vertices in the same set**.**



Example

A college has to fit French, geography, History, Maths and Science into a single timetable slot. There are five teachers available all of whom can teach two or more of these subjects.

Ann can teach French and Geography
Bob can teach French, Maths and Science
Carol can teach Geography and History
David can teach Geography, Maths and Science
Elaine can teach History Maths and Science

How should the college allocate the staff so that all subjects are covered?

Before the exam you should know:

- What is meant by a bipartite graph.
- That a matching maps vertices in one set to vertices in a second set. No vertex may be used more than once.
- For a complete matching the two sets must have the same number of vertices.
- A complete matching pairs every vertex in the first set to one in the second set.
- A maximal matching is one where there is no solution that uses a greater number of edges.
- A complete matching is always maximal, but a maximal matching is not necessarily complete.

The algorithm for finding a maximum matching

1. Always start with an initial matching

2. If the matching is not maximal it can be improved by finding an ***alternating path.***

An alternating path:

- Starts on an unmatched vertex on the right hand side
- Consists of edges alternately *not in* and *in* the matching
- Finishes on an unmatched vertex in the second set

3. If every vertex is now matched so we have a complete matching. If it is not, then repeat step 2.

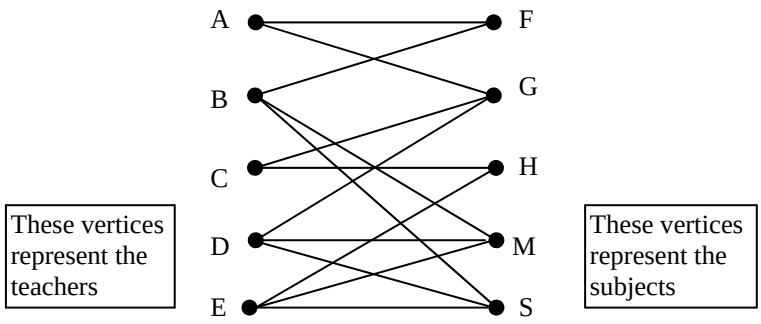
4. The solution consists of:

- edges ***in*** the alternating path but ***not in*** the initial matching;
- edges ***in*** the initial matching but ***not in*** the alternating path.

Disclaimer: Every effort has gone into ensuring the accuracy of this document. However, the FM Network can accept no responsibility for its content matching each specification exactly.

Solution

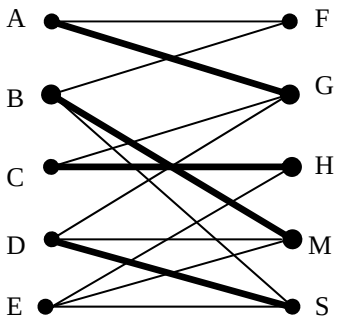
Start by drawing a bipartite graph to model the situation



Start with an initial matching:

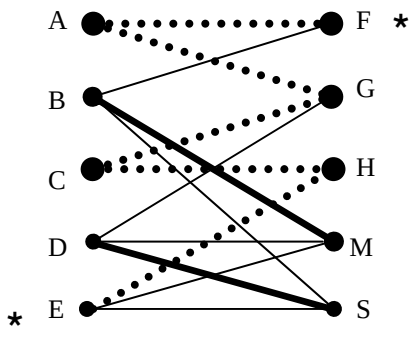
A – G, B – M, C – H, D – S

this is not a maximum matching since Elaine has not been allocated a subject and there is no-one to teach French.



We must try to find an ***alternating path***

Start on an unmatched vertex on the right hand side (F)
Choose an edge which is not in the initial matching (FA)
Choose an edge which is in the initial matching (AG)
Choose an edge which is not in the initial matching (GC)
Choose an edge which is in the initial matching (CH)
Choose an edge which is not in the initial matching (HE)



We have now reached E which was not in the initial matching so we have ***breakthrough.***

The laternating path is F – A – G – C – H – E

The solution consists of:

- edges ***in*** the alternating path but ***not in*** the initial matching: AF, CG, EH
- edges ***in*** the initial matching but ***not in*** the alternating path: BM, DS

so the solution is:

Ann teaches French
Bob teaches Maths
Carol teaches Geography
David teaches Science
Elaine teaches History

Disclaimer: Every effort has gone into ensuring the accuracy of this document. However, the FM Network can accept no responsibility for its content matching each specification exactly.

REVISION SHEET – DECISION MATHS

NETWORK FLOWS

The main ideas are covered in			
AQA	Edexcel	MEI	OCR
D2	D1		D2

The main ideas in this topic are:

- Modelling flows using bipartite graphs.
- Finding the maximum flow through a network.
- The maximum flow-minimum cut theorem

Before the exam you should know:

- What is meant by source, sink and capacity.
- What a cut is.
- How to find an initial flow.
- The meaning of a flow augmenting path and how to find them.
- How to use the labeling procedure.
- What is meant by excess capacity and back capacity.
- What is meant by a saturated arc.
- The maximum flow – minimum cut theorem.
- How to insert a super-source and super-sink into a network.

The algorithm for finding a maximum flow

1. Always start with an initial feasible flow, found by inspection.
2. Label each arc with
   - the flow along it, shown by an arrow pointing back towards the source
   - the excess capacity, which is the amount by which the flow could be increased, shown by an arrow pointing forward towards the sink
3. Systematically look for flow augmenting paths and mark these on your network using the labelling procedure
4. When all paths are blocked by saturated arcs you have found the maximum flow.

Example

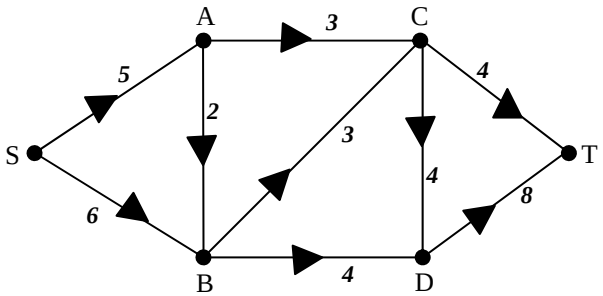
In this the directed network

a) What is the maximum flow along the path SACT?

b) Find an initial flow of value 7.

c) Find the maximum flow in the network

d) What are the capacities of these cuts



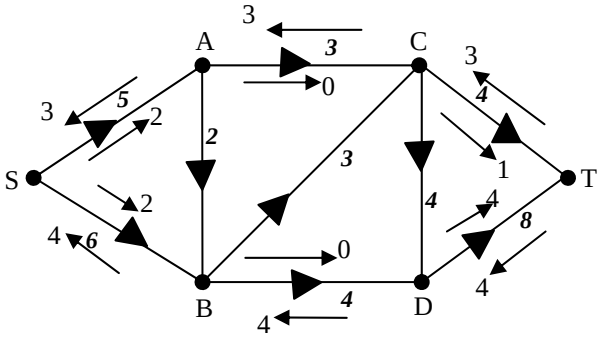
Solution

a) The maximum flow along SACT is 3
(this is determined by the arc of least capacity on the path).

b) A flow of 7 (shown on the diagram) is

SACT with capacity 3

SBDT with capacity 4



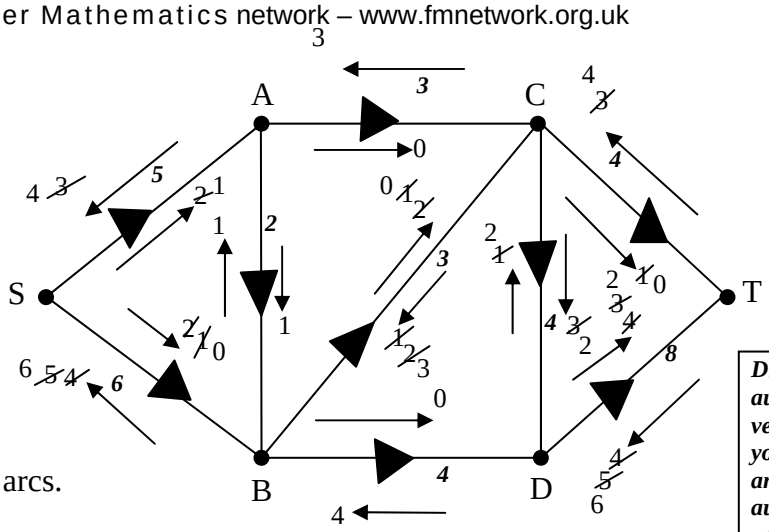
Disclaimer: Every effort has gone into ensuring the accuracy of this document. However, the FM Network can accept no responsibility for its content matching each specification exactly.

c) Flow augmenting paths

SBCT with capacity 1

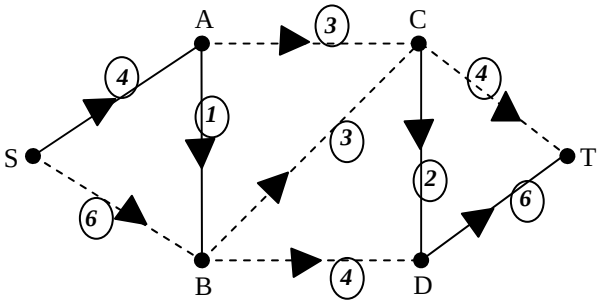
SBCDT with capacity 1

SABCDT with capacity 1



Diagrams showing the flow augmenting paths can be very messy. Try to keep yours as tidy as possible and always list the flow augmenting paths you have used

This gives a maximum total flow of 10. The flow is shown on this diagram, along with the saturated arcs.

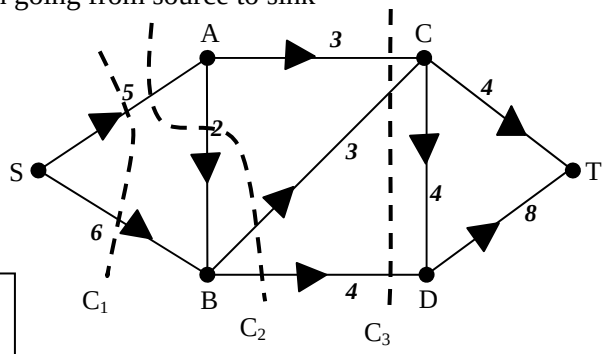


Cuts

A cut partitions the vertices into two sets, one containing the source and one containing the sink.
The capacity of a cut is the total of all the cut edges with direction going from source to sink

Find the capacity of the cuts shown on the directed network:

Note that only three cuts have been shown here, but there are many more cuts in this network.



C_1 is the cut {S}, {A, B, C, D, T}

It has capacity 5 + 6 = 11

C_2 is the cut {S, B}, {A, C, D, T}

It has capacity 5 + 0 + 3 + 4 = 12

Note that we do not add the capacity of arc AB as it is directed from the sink side of the cut to the source side

C_3 is the cut {S, A, B}, {C, D, T}

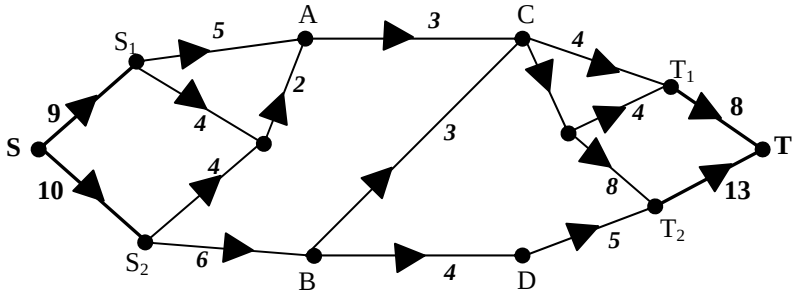
It has capacity 3 + 3 + 4 = 10

Maximum flow- minimum cut theorem.

The theorem states that the maximum flow in a directed network is equal to the capacity of the minimum cut.
In the example above the cut C_2 is the minimum cut and it has a value 10. This confirms that the flow of 10 found in (c) above is the maximum flow.

Networks with many sources and sinks

If there is more than one source (S_1 and S_2 on the diagram) or sink (T_1 and T_2 on the diagram) you must introduce ***supersource (S)*** and/or ***supersink (T).***



SS_1 must have a capacity 5 + 4 = 9
SS_2 must have capacity 4 + 6 = 10
T_1T must have capacity 4 + 4 = 8
T_2T must have capacity 8 + 5 = 13

Disclaimer: Every effort has gone into ensuring the accuracy of this document. However, the FM Network can accept no responsibility for its content matching each specification exactly.