Beckfoot

Further Maths | AS Discrete | Year 12

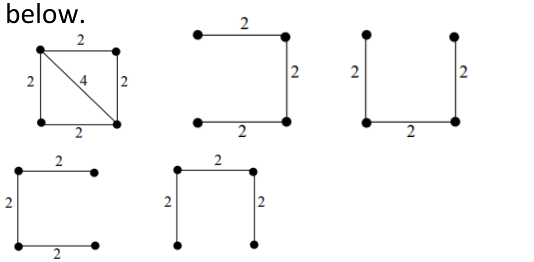
enjoy learn succeed

Graphs and Networks

1	Vertex/ Node	A vertex is a point on a graph that is connected to other vertices by edges.
2	Arc/ Edge	An edge is a line connecting two vertices on a graph.
3	Weight	A numerical value (often a distance) associated with an arc on a network.
4	Trail	A trail is a sequence of edges in which the end of one edge (except the last) is the beginning of the next, and no edge is repeated.
5	Cycle	A cycle is a closed path (i.e. the end of the last edge is the start of the first, and no vertices are repeated except that the final vertex is the same as the first).
6	Connected Graph	A graph is connected if a path exists between every pair of vertices.
7	Degree	The degree (or order, or valency) of a vertex is the number of ends of edges connecting into it.
8	Subgraph	A subgraph of a graph is a subset of the vertices together with a subset of the edges.
9	Subdivision	A subdivision of a graph occurs when a new vertex is added on an edge so that the edge becomes two edges. So if there is an edge AB, and you add a vertex P so that there is no longer an edge AB, but instead there are two edges AP and BP, then you have subdivided the graph.
10	Loop	An edge whose beginning and end both link into the same vertex.

11	Hamiltonian	A Hamiltonian graph is a graph for which a Hamiltonian cycle exists.
12	Complete Graph	A complete graph is a simple graph in which every pair of vertices is connected by an edge. A complete graph with n vertices is denoted as K_n.
13	Bipartite Graph	A bipartite graph is one in which the vertices fall into two sets, and each edge has a vertex from one set at one end, and a vertex from the other set at the other end. A bipartite graph with r vertices in one set and s vertices in the other, in which every vertex in each set is connected to each vertex in the other (a complete bipartite graph) is denoted as $K_{r,s}$
14	Adjacency Matrix	An adjacency matrix (incidence matrix) is a way of representing a graph on a matrix. Rows and columns correspond to vertices, and the entries in the matrix correspond to the number of edges from one vertex to another.
15	Simple Graph	A graph in which there are no loops and in which there is no more than one edge connecting any pair of vertices.
16	Tree	A tree is a graph in which all vertices are connected by edges, but there are no cycles (a cycle is a circuit of edges).
17	Euler's Formula	If v is the number of vertices in a connected planar graph, e is the number of edges, and f is the number of faces (i.e. distinct regions defined by edges, including the infinite outer one) then the following formula is satisfied: $v-e+f=2$. This is Euler's formula for connected planar graphs.
18	Connected Planar Graphs	A graph is planar if it can be drawn in two dimensions without any of the edges crossing.

Minimum Spanning Trees

1	**Make sure that you learn Kruskal's algorithm properly** You are expected to remember this algorithm and be fluent in its use. If you do the suggested questions, this shouldn't be a problem.
2	**Make sure that you learn Prim's algorithm properly** Again, you are expected to remember this algorithm and be fluent in its use. Also note that you need to be able to apply Prim's algorithm both directly to a network, and to an incidence matrix representing a network.
3	**Remember which algorithm is which!** Sometimes you may be told which algorithm to use, in which case it is essential that you use the right one! Otherwise, you may be asked to use a suitable algorithm and state its name – if you carry out one of the algorithms correctly but give the wrong name you will lose an easy mark.
4	**Remember that more than one minimal spanning tree may exist for a network.** E.g. this network has four possible MSTs, shown below. 

Beckfoot

Further Maths	AS Discrete	Year 12

enjoy learn succeed

Travelling Salesperson Problem

1	**Remember that the nearest neighbour algorithm does not need to start from the point where the route is to start** You can start the nearest neighbour algorithm from any point. This gives a tour, ending up at the starting point, which can be rewritten so that any point can be the starting point of the route.
2	**Remember that a direct link is not always the shortest route** Sometimes going via a vertex which is already in the tour produces a quicker route than a direct link. If the graph is not already complete, a matrix of shortest distances between all pairs of vertices should be drawn up. The nearest neighbour algorithm should be performed on a complete graph or a matrix of shortest distances – this is the 'classical problem' as opposed to the 'practical problem'.
3	**If you are asked to give a route make sure that you include all vertices visited, including ones already visited** The nearest neighbour algorithm gives you the order of visiting the vertices. However, in the practical problem (rather than the classical problem), in some cases this involves going via a vertex already visited. If you are asked for the route rather than just the order, make sure that you include all vertices visited.
4	**Remember that the greatest lower bound and the least upper bound are used**. The greatest (largest) lower bound and least (smallest) upper bound are nearest to the optimal solution.

The Route Inspection Problem

1	**Make sure you pair up odd vertices correctly** If there are more than four odd vertices, writing out all the possible pairings becomes very tedious. In such cases it is usually possible to spot some obvious pairings, and if there are four vertices left over the possible pairings can be checked easily.
2	**Remember that the direct route is not always the shortest** When pairing up odd vertices, check that you have looked at the shortest route for each pair. This may not always be the direct route!
3	**When looking for a suitable route, note which edges need to be traversed twice** There are usually many possible optimal routes for the route inspection problem. Make a list of the edges which need to be traversed twice (if the vertices pairings involve going via another vertex, write down all edges involved). This should help you to find a suitable route.

Old City of Konigsberg with 7 Bridges

Network Flows

1	**Flow**	A flow along an arc in a network with a single source and a single sink is a non-negative number assigned to that arc. The flow must satisfy the feasibility condition and the conservation condition.
2	**Capacity**	The capacity of an arc in a network is the upper limit to the flow along that arc.
3	**Make sure you understand what is meant by a cut and its capacity** Remember in particular that a cut must divide a network into two parts, one of which contains the source and the other the sink. A cut may include arcs which are directed from the part containing the sink to the part containing the source – remember that such arcs must be included if you are listing the arcs in the cut, but that you do not include the capacities of such arcs when finding the capacity of the cut.	
4	**Remember the maximum flow – minimum cut theorem** If you have checked all possible cuts and identified the minimum cut, this tells you the maximum flow for the network.	
5	**Remember that forward arcs in the minimum cut must be saturated**	
6	**Make sure that you can deal with multiple sources and sinks** Remember to add a supersource and supersink in networks with multiple sources and sinks.	

Beckfoot

Further Maths	AS Discrete	Year 12

enjoy learn succeed

Critical Path Analysis

1	**Make sure that you consider all preceding activities** You must consider all of the preceding activities allocating the earliest time to an activity, and you must consider all of the following activities before allocating the latest time for the activity. If you follow the algorithm for calculating early times and late times correctly, this should not be a problem.
2	**Remember that there may be more than one critical path** Any activity for which the late time minus the early time is equal to the duration of the activity is critical. Sometimes there may be branches in the critical path, so that more than one chain of critical activities can be traced from the start event to the finish event of the project. Note that a critical activity must be part of such a chain. If you think an activity is critical, but it is not part of a chain from the start event to the finish event of the project, you have made a mistake.

Linear Programming

1	**Make sure that you are fluent with plotting straight-line graphs** This is a skill you will need throughout your A level Maths and Further Maths work, so you need to make sure you have really mastered it.
2	**Define your variables carefully In some questions the variables will be defined for you, but in others you need to define them yourself**. It is important to be precise – for example “Let x be the energy drink” would earn no marks. You must write “Let x be the number of litres of the energy drink which are to be produced”
3	**Use graph paper Always use graph paper to plot your graph, otherwise you may not be able to read of the vertices with sufficient accuracy.**
4	**Make sure that you understand the terminology** Make sure that you are familiar with all the terms in the glossary.
5	**Be careful to shade out the region on the correct side of a constraint line on the graph of an LP problem** Check a point to see whether or not it satisfies the constraint. If it does, shade out the region on the other side of the constraint line to the point. If it doesn’t satisfy the constraint, shade out the region on the same side as the point. See the Notes and Examples for an example.

6	**Remember that the optimum solution to an IP problem is not always the integer solution closest to the LP solution** You will need to check the integer solutions close to the LP solution to see which is best. At this level of work this is the only method available to you. Checking the two or three points closest to the IP solution will be enough to find the optimum solution in the problems that you will meet.
7	**Make sure that the integer points you are considering are within the feasible region** It is not always possible to tell from a graph if an integer point is in the feasible region, especially if large numbers are involved. Check that the point you are considering satisfies all the constraints. A good approach is to take the integer values of x either side of the exact vertex, and substitute each of these into each of the constraints, to find the largest (or in the case of a minimisation problem, smallest) possible integer value of y which satisfies all the constraints.
8	**Be careful to formulate minimisation problems correctly** Read questions carefully. Minimisation problems often relate to minimising costs. The constraints will involve mostly ⍰⍰ or inequalities, whereas maximisation problems involve mostly ⍰⍰ or inequalities.

M
r

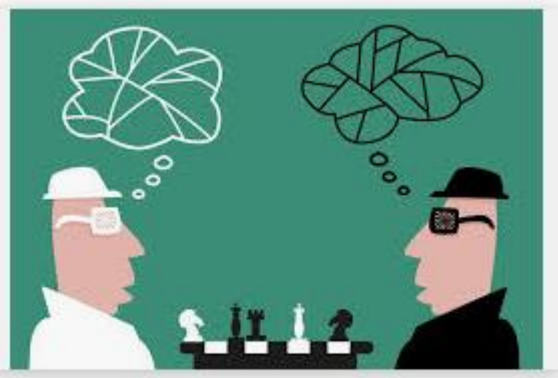
Beckfoot

Further Maths	AS Discrete	Year 12

enjoy learn succeed

Game Theory

1	**Make sure that you know how to find play-safe strategies** Remember that for the first player, you need to find the row minima and take the maximum of these (the maximin strategy), whereas for the second player, you need to find the column maxima and take the minimum of these (the minimax strategy). Make sure that you indicate all the row minima and column maxima.
2	**Check for stable solutions Remember that you should check for stable solutions before starting to look for the optimal mixed strategy.** In examination questions, you will often be asked to “show that the game has a stable solution” or “show that the game does not have a stable solution”. To do this, you need to clearly indicate all the row minima and column maxima, and state explicitly that the maximum of the row minima either is, or is not, equal to the minimum of the column maxima.
3	**Remember to look for dominance** If one or more row and / or column can be ignored due to dominance, the problem is simplified. For example, a 3 × 3 game would normally require linear programming methods, but if dominance allows you to ignore a row or column, you will be able to use the graphical method. Remember to check both rows and columns
4	**Draw graphs carefully and interpret them correctly** When using a graphical method for a 2 × n game, you do need to be fairly accurate with the graph, as otherwise it may not be clear which point you need to consider. Remember that your graph only need cover values of p between 0 and 1. Work out the expected pay-offs for p = 0 and p = 1, and use these to draw the lines. Remember that the shaded area is the area under ALL the lines, and that you need the highest point of the shaded area.
5	**Be careful if you need to work with a transposed pay-off matrix** To solve n × 2 games, you need to transpose the matrix. Remember that you must change the sign of all the outcomes, since you are now considering the game from the point of view of the second player. Remember that the value of the original game is the negative of the value of the transposed game.



Binary Operations

1	**Make sure you know the meanings of associative and commutative** For an associative operation □ acting on a set (a □ b) □ c = a □ (b □ c) for all a, b, c in the set For a commutative operation □ acting on a set a □ b = b □ a for all a, b in the set
2	**Make sure that you prove properties for all cases** If you are asked to prove that a given operation on a given set is commutative or associative, or that the set is closed, you must prove it for all possible cases. For a small, finite set, a Cayley table will allow you to check for some properties.