Curriculum Map Further Maths

Year 1	Core		Applied	
	Big Questions	Small Questions	Big Questions	Small Questions
Autumn 1	Complex Numbers	• Understand and use the definitions of imaginary and complex numbers. • Add and subtract complex numbers. • Multiply complex numbers. • Understand the definition of a complex conjugate. • Divide complex numbers. • Solve quadratic equations that have complex roots. • Solve cubic or quartic equations that have complex roots.	Algorithms and graph theory	• Use and understand an algorithm given in words. • Understand how flow charts can be used to describe algorithms. • Carry out a bubble sort. • Carry out a quick sort. • Carry out the three bin-packing algorithms and understand their strengths and weaknesses. • Determine the order of an algorithm. • Know how graphs and networks can be used to create mathematical models. • Be familiar with basic terminology used in graph theory. • Know some special types of graph. • Understand how graphs and networks can be represented using matrices.
	Matrices	• Understand the concept of a matrix. • Define the zero and identity matrices. • Add and subtract matrices. • Multiply a matrix by a scalar. • Multiply matrices. • Calculate the determinant of a matrix. • Find the inverse of a matrix. • Use matrices to solve systems of equations. • Interpret simultaneous equations graphically.	Allocation (Assignment) problems	• Reduce cost matrices. • Use the Hungarian algorithm to find a least cost allocation. • Adapt the Hungarian algorithm to use a dummy. • Modify the Hungarian algorithm to deal with a maximum profit allocation. • Adapt the Hungarian algorithm to manage incomplete data.

Autumn 2	Matrices	• Understand the properties of linear transformations and represent them using matrices. • Perform reflections and rotations using matrices. • Carry out enlargements and stretches using matrices. • Find the coordinates of invariant points and the equations of invariant lines. • Carry out successive transformations using matrix products. • Understand linear transformations in three dimensions. • Use inverse matrices to reverse linear transformations.	Algorithms on graphs I	• Use Kruskal’s algorithm to find a minimum spanning tree. • Use Prim’s algorithm on a network to find a minimum spanning tree. • Apply Prim’s algorithm to a distance matrix. • Use Dijkstra’s algorithm to find the shortest path between two vertices in a network. • Use Floyd’s algorithm.
	Complex Numbers	• Show complex numbers on an Argand diagram. • Find the modulus and argument of a complex number. • Write a complex number in modulus-argument form. • Represent loci on an Argand diagram. • Represent regions on an Argand diagram.	Flows in Networks	• Understand and analyse flow through a network. • Find initial flows in networks. • Use the labelling procedure to augment a flow to determine the maximum flow in a network. • Use the maximum flow – minimum cut theorem to prove that a flow is maximal.

Spring 1	Series	• Use standard results for $\sum_{r=1}^{n} 1$ and $\sum_{r=1}^{n} r$ • Use standard results for $\sum_{r=1}^{n} r^2$ and $\sum_{r=1}^{n} r^3$ • Evaluate and simplify series of the form $\sum_{r=m}^{n} f(r)$, where f(r) could be linear, quadratic or cubic.	Algorithms on graphs II	• Use the orders of nodes to determine whether a graph is Eulerian, semi-Eulerian or neither. • Use the route inspection (Chinese Postman) algorithm to find the shortest route in a network.
	Algebra and functions (Roots of polynomials)	• Derive and use the relationships between the roots of a quadratic equation. • Derive and use the relationships between the roots of a cubic equation.	Linear programming	• Formulate a problem as a linear programming problem. • Illustrate a two-variable linear programming problem graphically.

		• Derive and use the relationships between the roots of a quartic equation. • Evaluate expressions relating to the roots of polynomial equations. • Find the equation of a polynomial whose roots are a linear transformation of the roots of a given polynomial.		• Locate the optimal point in a feasible region using the objective line (ruler) method. • Use the vertex testing method to locate the optimal point. • Determine solutions that need integer values.
	Proof	• Understand the principle of proof by mathematical induction and prove results about sums of series. • Prove results about divisibility using induction. • Prove results about matrices using induction.		

Spring 2	Vectors	• Understand and use the vector and Cartesian forms of the equation of a straight line in three dimensions. • Understand and use the vector and Cartesian forms of the equation of a plane.	Critical Path Analysis	• Model a project by an activity network using a precedence table. • Use dummy activities. • Identify and calculate early and late event times in activity networks. • Identify critical activities.

		• Calculate the scalar product for two 3D vectors. • Calculate the angle between two vectors, two lines, a line and a plane, or two planes. • Understand and use the scalar product form of the equation of a plane. • Determine whether two lines meet and determine the point of intersection. • Calculate the perpendicular distance between: two lines, a point and a line, or a point and a plane.		• Calculate the total float of an activity. • Calculate and use Gantt (cascade) charts.
			Game Theory	• Understand two-person games and the pay-off matrix. • Determine play-safe strategies and stable solutions (saddle points). • Determine the optimal mixed strategy for a game with no stable solution, for the player with two choices in a 2 x 3, 3 x 2, 2 x 4 or 4 x 2 game.

1 Sum	Calculus (Volumes of revolution)	• Find the volume of revolution when a curve is rotated about the x-axis. • Find the volume of revolution when a curve is rotated about the y-axis.	Recurrence Relations	• Use recurrence relations to describe sequences and model situations. • Find solutions to first order recurrence relations.

Summer 2	Complex Numbers	• Express a complex number in exponential form. • Multiply and divide complex numbers in exponential form. • Understand de Moivre’s theorem. • Use de Moivre’s theorem to derive trigonometric identities. • Use de Moivre’s theorem to find sums of series. • Know how to solve completely equations of the form z^n – a – ib = 0, giving special attention to cases where a = 1 and b = 0 • Use complex roots of unity to solve geometric problems.	Algorithms and graph theory.	• Use the planarity algorithm to determine whether or not a given graph is planar.
			Transportation problems	• Describe and model transportation problems. • Use the north-west corner method. • Understand unbalanced transportation problems and degenerate solutions. • Use shadow costs to find improvement indices. • Use the stepping-stone method. • Formulate a transportation problem as a linear programming problem.

Year 2	Core		Applied	
	Big Questions	Small Questions	Big Questions	Small Questions
Autumn 1	Hyperbolic functions	• Understand the definitions of hyperbolic functions. • Sketch the graphs of hyperbolic functions. • Understand and use the inverse hyperbolic functions. • Prove identities and solve equations using hyperbolic functions. • Differentiate and integrate hyperbolic functions.	Algorithms on graphs II	• Use the route inspection algorithm in networks with more than four odd nodes. • Explain the differences between the classical and practical problems. • Use a minimum spanning tree method to find an upper bound. • Use a minimum spanning tree method to find a lower bound. • Use the nearest neighbour algorithm to find an upper bound.
	Polar coordinates	• Understand and use polar coordinates. • Convert between polar and Cartesian coordinates. • Sketch curves with r given a s a function of Ө.	Allocation (Assignment) problems	• Formulate allocation problems as linear programming problems.
			Flows in Networks	• Analyse flows through a network that includes lower capacities. • Solve problems involving multiple sources and sinks. • Adapt solutions to deal with nodes of restricted capacity.

Autumn 2	Polar coordinates	• Find the area enclosed by a polar curve. • Find tangents parallel to, or at right angles to, the initial line.	Linear programming	• Understand and use slack and surplus variables. • Solve maximising and minimising linear programming problems using simplex tableaux. • Use the simplex tableau method to solve linear programming problems requiring integer solutions. • Understand and use the two-stage simplex method for maximising and
	Series	• Understand and use the method of differences to sum finite series. • Find and use higher derivatives of functions. • Know how to express functions as an infinite series in ascending powers using Maclaurin series expansion.		

		• Be able to find the series expansions of compound functions.		minimising problems which may include $\ll$ and $\gg$ constraints. • Understand and use the Big-M method

Spring 1	Further calculus	• Evaluate improper integrals. • Understand and evaluate the mean value of a function. • Integrate rational functions using trigonometric substitutions. • Integrate using partial fractions.	Dynamic Programming	• Understand the terminology and principles of dynamic programming, including Bellman’s principle of optimality. • Use dynamic programming to solve maximum, minimum, minimax or maximin problems, presented in network form or table form.
			Decision Analysis	• Use, construct and interpret decision trees. • Calculate expected monetary values (EMVs). • Use utility to compare different courses of action.

Spring 2	Further volumes of revolution	• Find volumes of revolution around the x-axis. • Find volumes of revolution around the y-axis. • Find volumes of revolution for curves defined parametrically. • Model real-life applications of volumes of revolution.	Critical Path Analysis	• Construct resource histograms. • Construct scheduling diagrams.
	Methods in differential equations	• Solve first-order differential equations using an integrating factor. • Solve second-order homogeneous differential equations using the auxiliary	Game Theory	• Reduce a pay-off matrix using dominance arguments. • Convert games into linear programming problems.

		equation. • Solve second-order non-homogeneous differential equations using the complimentary function and the particular integral. • Find particular solutions to differential equations using given boundary conditions.		

Summ	Modelling with differential equations	• Model real-life situations with first-order differential equations. • Use differential equations to model simple harmonic motion.	Recurrence relations	• Find solutions to second order recurrence relations.