In this session we will examine in detail the rubrics and requirements of the school-assessed task, SAT assessments for Unit 4 VCE Algorithmics in 2021.

My name is Georgia Gouros, and I am the State Reviewer for VCE Algorithmics and presenter of this session.

The copyright in this PowerPoint presentation is owned by the Victorian Curriculum and Assessment Authority or in the case of some materials, by third parties. No part may be reproduced by any process except in accordance with the provisions of the Copyright Act of 1968 or with permission from the Copyright Officer at the Victorian Curriculum and Assessment Authority.

In this session, Unit 4 SAT Criteria 1–6 will be covered in addition to authentication and assessment.

Unit 4 SAT overview. The Unit 4 School-assessed Task consists of three components over the three outcomes. These components are: two written explanations, an algorithm design task in two parts, and an explanation of the universality of computation and algorithms. The Unit 4 School-assessed Tasks consists of those three components over the three outcomes as discussed. The two written explanations an algorithm design task in two parts, an explanation of the universality of computation and algorithms assessed against six criteria: understanding of formal algorithm analysis, skills in establishing the efficiency of a naive algorithm, understanding of advanced algorithm design, skills in developing an improved algorithm in response to a naive algorithm. Understanding the principles of computation and skills in the discussion and evaluation of computation concepts.

The table summarises the outcome components against the rubric criteria for Unit 4 SAT tasks. Outcome 1 pertains to the study of formal algorithm analysis and is assessed against Criteria 1 by timed written explanation test. And Criteria 2, by an algorithm design Task 1. Outcome 2 pertains to the study of advanced algorithm design and is assessed against Criteria 3 by a timed written explanation test. And Criteria 4, by algorithm design Task 2. Outcome 3 pertains to the study of universality of computation and algorithms and is assessed against Criteria 5 and 6 by written reports. Written explanation

Timed and supervised tests. Two components: formal analysis techniques and practical limits of computability. Algorithm design patterns and techniques for addressing the limits of computation. Each completed under test conditions within a 45 to 60 minute timeframe assessed against Criteria 1 and 3. The task must allow for students to demonstrate the full range of performance. A mixture of question types is appropriate.

Algorithm design. The design of an algorithm, consisting of two components. Part 1, a formal analysis of a given naive algorithm of approximately 400 words. And part two, a response to a naive algorithm consisting of an improved algorithm design, an analysis of the improved design including its correctness assessed against Criterion 2 and 4. The naive algorithm should have scope for improvement through the application of one of the algorithms design patterns studied in Outcome 2. It should take a straightforward approach to the problem and not be unnecessarily complicated.

Explanation. Explanation of the universality of computation and algorithms. An explanation of the universality of computation and algorithms in one or more of the following forms: a written report approximately 700–800 words a visual report or an oral report of approximately 10 to 15 minutes. Students are to develop a report using core concepts from theoretical computer science studied in class. They should be able to give detailed descriptions, explanations and evaluations of these core concepts. This is assessed against Criteria 5 and 6. Consider giving your students a choice of mode of report and considerations with authentication.

Unpacking the criteria. In the next section, we will look at how each of the Unit 4 outcomes key knowledge and key skills lead to the criteria.

Written explanation Ensure a good spread of difficulty in questions. These tests should allow for questions to cover the full range of performance as described in the criteria.

Written explanation A written explanation of Outcome 1, Criterion 1, formal analysis techniques and the practical limits of computability. Outcome 2, Criterion 3, algorithm design patterns and techniques for addressing the limits of computation. Each task is to be completed under test conditions within a 45 to 60 minute timeframe. Each task must allow for students to demonstrate the full range of performance. A mixture of question types is appropriate.

Criterion 1 Test Key knowledge: the concept of algorithm complexity including time and space complexity. Student outcomes, very high, 9–10 score. Compares the limits placed on the use of an algorithm by its time complexity with those placed by its space complexity. Teachers must reference the criteria descriptors to set questions to assess the level of knowledge and skills from very low to very high.

Criteria 1 Test Key knowledge: the concept of P and NP-complete complexity classes. Students scores, 9–10, very high. Descriptors: Discusses the practical consequences of a problem belonging to either the P or NP-Complete complexity class.

Criteria 1 Test Key knowledge: the concepts of Big-O, Big-Ω and Big-θ notation, the differences between best case and worst case complexity analysis of algorithms, The P time complexity class and examples of algorithms that have complexities of
O , O , O , O , O , O . Teachers must reference the criteria descriptors to set test questions to assess the level of knowledge and skill from very low to very high. For example, student scores 9–10, very high. Students descriptor: Explains the algorithmic structures that give rise to O , O , O and O time complexities.

Criterion 1 Test Key knowledge includes: an understanding that problems can be harder than P problems and the implications for their solvability, consequences of combinatorial explosions and indicators for them. demonstrates how exponentially sized search and solution spaces impose practical limits on computability. Students scores of 9–10, very high. Student scores of 7–8, high. Students demonstrate clearly how combinatorial algorithms create exponentially sized search spaces.

Criterion 1 Test Key knowledge: recurrence relations as a method of describing the time complexity of recursive algorithms, the Master Theorem for solving recurrence relations of the form, T as a function of n equals a times T as a function of n divided by b plus k times n to the power of c. Teachers must reference the criteria descriptors again as before to set test questions to assess the level of knowledge from very low to very high. For example, student scores 9–10, very high: Explains the connections between the call trees produced by recursive functions and the solutions of the three cases of the Master Theorem.

Criteria 3 Test Key knowledge: the divide and conquer algorithms that have linear time split and merge steps, including mergesort and quicksort, dynamic programming algorithms that require no more than a single dimension array for storage, including Fibonacci numbers, the one dimensional knapsack problem and change making problems, tree search by backtracking and its applications, differences between divide and conquer and dynamic programming and their applications, induction and contradiction as methods for demonstrating the correctness of dynamic programming and backtracking algorithms, the minimax method to solve combinatorial problems. Key skills: recognise and apply the divide and conquer dynamic programming and backtracking design patterns, apply the minimax method, propose an argument for the correctness of an algorithm.

Criteria 3 Test Teachers must reference the criteria descriptors to set test questions to assess the level of knowledge and skill from very low to very high. Students scores 9–-10, very high. Descriptors include: Proposes a valid argument for the correctness of one of the specified divide and conquer or dynamic programming algorithms using either the induction or contradiction method.

Criteria 3 Test Key knowledge includes: heuristics and randomised meta-heuristic algorithms, including simulated annealing, as approaches to overcome soft limits of computation, limitations of heuristics and randomised meta-heuristic algorithms, the graph colouring, the one dimensional knapsack and travelling salesman problems and heuristic methods for solving them. Key skills include: uses heuristics to solve computationally hard problems, explain the meaning of randomised approaches for intractable problems, including the graph colouring, the one dimensional knapsack and travelling salesman problems.

Criteria 3 Test Design of an algorithm. The design of an algorithm, consisting of two components: Part 1, Outcome 1 Criteria 2. Formal analysis of a given naive algorithm of approximately 400 words. Part 2, Outcome 2, Criteria 4. A response to a naive algorithm consisting of an improved algorithm design, an analysis of the improved design including its correctness, approximately 600 words. This is assessed through two criteria. skills in establishing the efficiency of a naive algorithm and skills in developing an improved algorithm in response to a naive algorithm.

Criteria 2 Algorithm Design Key skills to be assessed include: formally analyse the efficiency of algorithms using Big-O notation, read off a recurrence relation for the running time of a recursive algorithm that can be solved by the Master Theorem that takes the form of: T as a function of n, which is equal to the summation form i equals 1 to k of T n minus a subscript i brackets plus b, where a subscript i is a natural number, uses the stated Master Theorem to solve recurrence relations. Teachers must reference the criteria descriptors to set a task that allows students to demonstrate this skill from very low to very high. For example, student scores of
9–10, very high, have the descriptor: Analyses precisely and elegantly the time complexity of the naive algorithm through the efficient selection and application of appropriate techniques.

Criteria 2 Algorithm Design Key skills include: identify input patterns that lead to best and worst case performance. As for before, teachers must reference the criteria descriptors. For example, student scores 9–10, very high. The descriptor is: Describes efficiently and precisely the class of input instances for the naive algorithm that would result in a best case and the worst case running times.

Criteria 4 Algorithm Design Key skills also include: develop and compare different algorithms for solving the same problem, using different design patterns. Again, student scores 9–10, very high. The descriptor: Describes efficiently, proposes, sorry. Proposes a valid argument for the correctness of the algorithm, if appropriate using either the induction or contradiction method. Criteria 4 Algorithm Design Key skills also include: develop and compare different algorithms to solving the same problem, using different algorithm design patterns. For example, student scores 7–8, high, Compares comprehensively and accurately the student-design algorithm and the naive algorithm for the problem highlighting the improvements made and any trade-offs required.

Criteria 4 Algorithm Design Descriptors continued here: Explanation of the universality of computation. An explanation of the universality of computation and algorithms is in one or more of the following forms: a written report, approximately 700 to 800 words, a visual report, an oral report of approximately 10–15 minutes. This is assessed through two criteria. understanding the principles of computation, skills in the discussion and evaluation of computation concepts. These criteria provide a broad range of performance descriptors so that a judgment of the student's level of performance can be made regardless of the topics explored by a student in their report. Students are not expected to demonstrate all of the criteria statements.

Criteria 5 Explanation Key knowledge includes: David Hilbert's 1927 program to fully formalised mathematical reasoning, including its goals, historical context, outcome and connection to the origin of computer science, characteristics of a Turing machine, implications of undecidability for Hilbert's program. Again, teachers must reference the criteria descriptors to set a task that allows students to demonstrate this skill from very low to very high. For example, student scores 9–10, very high. Descriptor: Explains comprehensively and precisely the operation of a Turing machine.

Criteria 5 Explanation Also includes as key knowledge: the concept of decidability and undecidability, including the Halting Problem and the demonstration that it is undecidable, and implications for automatic program verification, the meaning of the Church-Turing thesis and its limitations. Again, teachers must reference the criteria for setting tasks. Student scores of 9–10, very high, descriptor: Explains comprehensively and precisely conceptually how the equivalence of computational formalism is shown.

Criteria 5 Explanation Key knowledge includes: John Searle's Chinese Room Argument, including historical context, connections to artificial intelligence, standard responses for and against, characteristics of alternative modes of computation. See the 2021, VCAA guide online for 2021. This is DNA computing and neural-networks. Teachers must reference the criteria descriptors to set tasks, to allow students to demonstrate their skill from very low to very high. For example, student scores of
9–10, very high. Descriptor: Explains comprehensively and precisely an alternative model of computation, including significant technical details of the method and its operation.

Criteria 6 Explanation Key skills include: demonstrates the existence of hard limits computability, discuss the merit and limitations of the Church-Turing and Cobham theses, define decidability and undecidability and explain selected examples. Student scores of 9–10, very high. Descriptors include: Demonstrates precisely the undecidability of the Halting Problem through the use of comprehensive and well-structured argument.

Criteria 6 Explanation Key knowledge includes: the relationship between Turing machines and computational complexity theory the meaning of Cobham's thesis for computational complexity theory. Key skills include: explain conceptually how the equivalence of computational formalisms is shown. For example, student scores of 9–10, very high. Descriptor: Comprehensively and precisely evaluates the merits and limitations of Cobham's thesis and discusses it's meaning for computational complexity theory.

Criterion 6 Explanation Key skills include: present and evaluate responses to the Chinese Room Argument, explain how alternative methods of computation might be used to overcome current limits of computation. Student scores of 9–10, very high descriptor: Explains comprehensively and precisely how an alternative method of computation might be used, and evaluate its merits and limitations.

The next session we will cover the authentication process for students SAT tasks. In order to establish the authentication of student work, student progress needs to be recorded by the teacher with dates and comments. Teachers need to regularly observe student work during the life of the SAT. Some teachers use one class a week to do this formally. Teachers must use the VCAA Authentication Record Form and update with comments from observations as you go. These can be requested as part of the audit process at a later time. Both you and the student are to sign and date your initials for each observation at the time of observation and for submission.

The next section we'll cover the assessment process for students SAT tasks. The VCE Algorithmics Assessment Sheet for Unit 4 assists teachers to determine the score for each student on completion of a SAT. Teachers need to make judgements on the student performance against each criteria by a score in the range of 0–10.

Any inquiries regarding the SAT assessment tasks for Unit 4 can be made by contacting, Phil Feain, Digital Technologies Curriculum Manager, VCAA by the phone, 9059-5146. Or by email: Phillip.Fean@education.vic.gov.au.

Thank you for watching this presentation.

[Copyright Victorian Curriculum and Assessment Authority 2020](https://www.vcaa.vic.edu.au/Footer/Pages/Copyright.aspx)