VCE Algorithmics (HESS) 2023-2026 Implementation on-demand video

Video 4
Background to the SAT

VICTORIAN CURRICULUM AND ASSESSMENT AUTHORITY

VICTORIA State Government

Acknowledgement of Country

The VCAA respectfully acknowledges the Traditional Owners of Country throughout Victoria and pays respect to the ongoing living cultures of First Peoples.

VCE Algorithmics (HESS) 2023-2026 Implementation on-demand video

Video 4
Background to the SAT

VICTORIAN CURRICULUM AND ASSESSMENT AUTHORITY

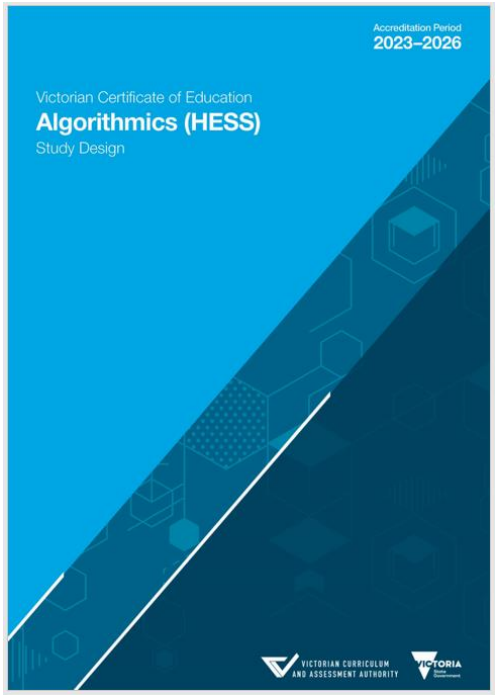
VICTORIA State Government

Purpose of this presentation

This presentation will cover:

- Unit 3 Outcome 3 SAT – Part 1
  - Key knowledge, key skills and the assessment task
- Unit 4 Outcome 1 SAT – Part 2
  - Key knowledge, key skills and the assessment task
- Unit 4 Outcome 2 SAT – Part 3
  - Key knowledge, key skills and the assessment task
- Advice for teachers

Algorithmics (HESS) and the SAT



- This is the Draft VCE Algorithmics (HESS) Study Design.
- The School-assessed Task now involves Unit 3 Outcome 3 as well as Unit 4 Outcomes 1 and 2.
- An all new Advice for teachers is being developed to support the study in 2023.
- An all new Administrative information for School-based Assessment will be published at the start of 2023.

The School-assessed Task

The SAT – Part 1

Unit 3 Area of Study 3 – Applied algorithms

In this area of study, students combine their knowledge of data modelling and algorithm design to solve real-world problems. Students consider a variety of algorithms and ADTs before selecting a suitable combination. They justify their chosen combination of algorithms and data types relative to other possible choices. Typically the fitness of a chosen combination could be measured in terms of the selection of salient features to achieve an appropriate level of abstraction and the quality of result produced by the algorithm.

The SAT – Part 1

Outcome 3

On completion of this unit the student should be able to design suitable solutions for real-world problems that require the integration of algorithms and data types, including the communication of solutions and their justification.

Unit 3 Outcome 3 – Key knowledge

Key knowledge

- characteristics and applicability of ADTs and algorithm design patterns
- suitability of ADTs and algorithm design patterns for a variety of problem contexts
- combinations of ADTs to meet complex problem requirements
- the application of algorithms to answering real-world problems

Unit 3 Outcome 3 – Key skills

Key skills

- describe how complex information can be represented by a combination of ADTs
- select combinations of ADTs and algorithms that are fit for purpose
- justify the suitability of ADTs and algorithm design patterns for particular problems
- communicate the design of data models and algorithms
- explain the interpretation of computed solutions in terms of their meaning to the original real-world problem being solved

Unit 3 Outcome 3 SAT – Part 1

Unit 3 Outcome 3	Assessment task
Design suitable solutions for real-world problems that require the integration of algorithms and data types, including the communication of solutions and their justification.	The design of a data model and algorithm combination to solve a real-world/applied problem, including: • a specification of the problem • a consideration of multiple solution options • the selection of a suitable, coherent, clear and fit-for-purpose solution

School-assessed Task – Part 1 for Unit 3 Outcome 3 will contribute 8 per cent to the study score.

The SAT – Part 2

Formal algorithm analysis

In this area of study, students investigate the efficiency of algorithms using mathematical techniques. Students learn how some computable problems require such a large amount of resources that in practice it is not possible to solve these exactly for realistic problem sizes. Students examine specific, widely occurring instances of such problems and the reasons why these problems cannot be solved. Students analyse time complexity formally and informally, while they study space complexity as a general concept. Students are not expected to derive the space complexity of algorithms.

The SAT – Part 2

Outcome 1

On completion of this unit the student should be able to establish the efficiency of simple algorithms and explain soft limits of computability.

Unit 4 Outcome 1 – Key knowledge

Key knowledge

- the concept of classifying algorithms based on their time and space complexity with respect to their input
- techniques for determining the time complexity of iterative algorithms
- the definition of Big-O notation and its application to the worst-case time complexity analysis of algorithms
- recurrence relations as a method of describing the time complexity of recursive algorithms
- the Master Theorem for solving recurrence relations of the form:

$$T(n) = \begin{cases} a \cdot T\left(\frac{n}{b}\right) + kn^c & \text{if } n > 1 \\ d & \text{if } n = 0 \end{cases}$$

where $a > 0, b > 1, c \geq 0, d \geq 0, k > 0$

and its solution: $T(n) = \begin{cases} O(n^c) & \text{if } a < b^c \\ O(n^c \log(n)) & \text{if } a = b^c \\ O\left(n^{\log_b(a)}\right) & \text{if } a > b^c \end{cases}$

- examples and common features of algorithms that have time complexities of $O(1)$, $O(\log n)$, $O(n)$, $O(n \log n)$, $O(n^2)$, $O(n^3)$, $O(2^n)$ and $O(n!)$
- the concept of the P, NP, NP-Hard and NP-Complete time complexity classes for problems
- consequences of combinatorial explosions and indicators for them
- the feasibility of NP-Hard problems in real-world contexts

Unit 4 Outcome 1 – Key skills

Key skills

- formally analyse the time 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 or takes the form: $T(n) = \sum_{i=1}^{k} T(n - a_i) + b$, where $a_i \in N$
- use the stated Master Theorem to solve recurrence relations
- demonstrate how exponentially sized search and solution spaces impose practical limits on computability
- evaluate the suitability of algorithms to particular contexts based on their time or space complexity
- estimate the time complexity of an algorithm by recognising features that are common to algorithms with particular time complexities
- describe characteristics of problems in the P, NP, NP-Hard or NP-Complete time complexity classes, including the consequences for a problem's feasibility of it belonging to one of these classes

Unit 4 Outcome 1 SAT – Part 2

Unit 4 Outcome 1	Assessment task
Establish the efficiency of simple algorithms and explain soft limits of computability.	A formal time complexity analysis of the designed algorithm for the applied problem and an explanation of the consequences of these results on the algorithm's real-world application.

School-assessed Task – Part 2 for Unit 4 Outcome 1 will contribute 6 per cent to the study score.

The SAT – Part 3

Advanced algorithm design

In this area of study, students examine more advanced algorithm design patterns. Students learn how to select algorithmic approaches from a wider range of options, depending on the structure of the problem that is being addressed. They investigate how some problems are solvable in principle while being intractable in practice. They explore examples of such problems with real-world relevance and learn how such problems can be tackled by computing near-optimal solutions.

The SAT – Part 3

Outcome 2

On completion of this unit the student should be able to solve a variety of information problems using algorithm design patterns and explain how heuristics can address the intractability of problems.

Unit 4 Outcome 2 – Key knowledge

Key knowledge

- the binary search algorithm
- divide and conquer algorithms that have linear time divide and merge steps, including mergesort and quicksort
- dynamic programming algorithms that require no more than a single dimension array for storage, including the Fibonacci numbers and change-making problem
- tree search by backtracking and its applications
- the application of heuristics and randomised search to overcoming the soft limits of computation, including the limitations of these methods
- hill climbing on heuristic functions, the A* algorithm and the simulated annealing algorithm
- the graph colouring, 0-1 knapsack and travelling salesman problems and heuristic methods for solving them

Unit 4 Outcome 2 – Key skills

Key skills

- apply the divide and conquer, dynamic programming and backtracking design patterns to design algorithms and recognise their usage within given algorithms
- develop different algorithms for solving the same problem, using different algorithm design patterns, and compare their suitability for a particular application
- apply heuristics methods to design algorithms to solve computationally hard problems
- explain the application of heuristics and randomised search approaches to intractable problems, including the graph colouring, 0-1 knapsack and travelling salesman problems

Unit 4 Outcome 2 SAT – Part 2

Unit 4 Outcome 2	Assessment task
Solve a variety of information problems using algorithm design patterns and explain how heuristics can address the intractability of problems.	A design of an improved data model and algorithm combination to solve the applied problem, including: • the selection of an efficient, coherent and fit-for-purpose solution • a time complexity analysis • a comparison to the original solution.

School-assessed Task – Part 3 for Unit 4 Outcome 2 will contribute 6 per cent to the study score.

Advice for teachers

Advice for teachers

- Overview of Unit 3: Algorithmic problem solving
- Teaching and learning activities
- Detailed examples
- School-assessed Task
  - Unit 3 Outcome 3
- Unit 3 Sample weekly planner

- Overview of Unit 4: Principles of algorithmics
- Teaching and learning activities
- Detailed examples
- School-assessed Task
  - Unit 4 Outcome 1
  - Unit 4 Outcome 2
- Unit 4 Sample weekly planner

Contact

- **Phil Feain – Digital Technologies Curriculum Manager (VCAA)**
- **Ph: (03) 9059 5146**
- **Philip.Feain@education.vic.gov.au**

Authorised and published by the
Victorian Curriculum and Assessment Authority