### **Analysis of Algorithms**

### **Duration: 3 Weeks**

### **Summary**

This lesson teaches about computing algorithms. It uses the first part of the Khan Academy readings and interactive exercises on algorithms. It shows the student classic computing algorithm steps such as sequencing, selection, iteration, and recursion. It illustrates classic computing algorithms such as searching and sorting. It has the student evaluate the time efficiency of algorithms. The lesson also shows that computing algorithms can be expressed using techniques like pseudocode and implemented by programming them in a programming language.

The assignment has the student add statements to Khan Javascript implementations of searching and sorting algorithms to see a visual representation of their efficiency in terms of the number of operations they perform. They are to run the program with data that generates a best case, average case, and worst case execution. It requires an understanding of the Khan Javascript programming that they have learned so far and an understanding of where operations are performed in algorithms.

Note that this module is not required for URI credit, but the material is on the AP Exam. Thus, there is not URI practical assessment, but we recommend that you check (and grade) that they did the Khan material.

The Professional Development video tutorial on how to teach this Algorithms module given by Dr. Fay-Wolfe is [here](https://youtu.be/JcGqmgCq-qE) [47:30min]

### **Learning Objectives**

* Define *algorithm*.
* Develop an algorithm for implementation in a program. [AP CSP P2, 4.1.1]
* Express an algorithm in a language (e.g. natural language, pseudocode, or programming language). [AP CSP P5, 4.1.2]
* Explain the difference between algorithms that run in a reasonable time and those that do not run in a reasonable time. [AP CSP P1, LO 4.2.1]
* Explain how different algorithms that solve the same problem can have different efficiencies. Illustrate this with sorting and searching algorithms.
* Explain the difference between solvable and unsolvable problems in computer science. [AP CSP P1, LO 4.2.2]
* Explain the existence of undecidable problems in computer science. [AP CSP P1, LO 4.2.3]
* Evaluate algorithms analytically and empirically for efficiency, correctness, and clarity. [AP CSP P4, LO 4.2.4]
* Explain how programs implement algorithms. [AP CSP P2, LO 5.2.1]

### **Course Material**

* **Algorithms Intro**
  + Watch: [What is an Algorithm and why should you care?](https://www.khanacademy.org/computing/computer-science/algorithms/intro-to-algorithms/v/what-are-algorithms) [4:39]
  + Read: [A Guessing Game](https://www.khanacademy.org/computing/computer-science/algorithms/intro-to-algorithms/a/a-guessing-game)
  + Read: [Route Finding](https://www.khanacademy.org/computing/computer-science/algorithms/intro-to-algorithms/a/route-finding)
* **Searching Algorithms**
  + Read: [Binary Search](https://www.khanacademy.org/computing/computer-science/algorithms/binary-search/a/binary-search)
  + Read: [Implementing Binary Search of an Array](https://www.khanacademy.org/computing/computer-science/algorithms/binary-search/a/implementing-binary-search-of-an-array)
  + Do Challenge: [Binary Search](https://www.khanacademy.org/computing/computer-science/algorithms/binary-search/p/challenge-binary-search)
  + Read: [Running Time of Binary Search](https://www.khanacademy.org/computing/computer-science/algorithms/binary-search/a/running-time-of-binary-search)
  + Self Assessment: [Running Time of Binary Search](https://www.khanacademy.org/computing/computer-science/algorithms/binary-search/e/running-time-of-binary-search)
  + Watch: [Analyzing Search Algorithms](https://www.youtube.com/watch?v=ysH2rtU1Q9A&feature=youtu.be) [9:48]
* **Sorting Algorithms**
  + Read: [Sorting](https://www.khanacademy.org/computing/computer-science/algorithms/sorting-algorithms/a/sorting)
  + Do Challenge: [implement swap](https://www.khanacademy.org/computing/computer-science/algorithms/sorting-algorithms/p/challenge-implement-swap)
  + Read: [Selection sort pseudocode](https://www.khanacademy.org/computing/computer-science/algorithms/sorting-algorithms/a/selection-sort-pseudocode)
  + Do Challenge: [Find minimum in subarray](https://www.khanacademy.org/computing/computer-science/algorithms/sorting-algorithms/p/challenge-find-minimum-in-subarray)
  + Do Challenge: [implement selection sort](https://www.khanacademy.org/computing/computer-science/algorithms/sorting-algorithms/p/challenge-implement-selection-sort)
  + Read: [Analysis of selection sort](https://www.khanacademy.org/computing/computer-science/algorithms/sorting-algorithms/a/analysis-of-selection-sort)
  + Read: [Insertion sort](https://www.khanacademy.org/computing/computer-science/algorithms/insertion-sort/a/insertion-sort)
  + **Optional:** Do Challenge: [implement insert](https://www.khanacademy.org/computing/computer-science/algorithms/insertion-sort/p/challenge-implement-insert).
  + Read: [Insertion sort pseudocode](https://www.khanacademy.org/computing/computer-science/algorithms/insertion-sort/a/insertion-sort-pseudocode)
  + **Optional:** Do Challenge: [Implement insertion sort](https://www.khanacademy.org/computing/computer-science/algorithms/insertion-sort/p/challenge-implement-insertion-sort)
  + Read: [Analysis of insertion sort](https://www.khanacademy.org/computing/computer-science/algorithms/insertion-sort/a/analysis-of-insertion-sort)
* **Undecidable and Unsolvable Problems**
  + Watch: [Limitations of Algorithms](https://www.youtube.com/watch?v=C6_I6EwW-Gw) [11:18]
  + Watch: [Three Problems Computers Can Never Solve](https://www.youtube.com/watch?v=u58DY1uKo5o) [3:01]
  + Watch: [Limits of Algorithms](https://www.youtube.com/watch?v=gBJ2leGs4Gw&feature=youtu.be) [3:17]
  + Read: [The Questions That Computers Can Never Answer](https://www.wired.com/2014/02/halting-problem/)
* **Algorithms Summary**
  + Read: [Algorithms](http://computing-concepts.cs.uri.edu/index.php/Algorithms)

Optional: For more on algorithms, see remaining Khan lessons on [Recursive algorithms, Towers of Hanoi, Merge Sort, Quick Sort, Graph Representation, Breadth-First Search](https://www.khanacademy.org/computing/computer-science/algorithms)

### **In Class**

* Optional: [Searching Algorithms Unplugged](http://csunplugged.org/searching-algorithms/)
* Optional: [Sorting Algorithms Unplugged](http://csunplugged.org/sorting-algorithms/)

### **Assessments**

* Conceptual Quiz:
  + [Analysis of Algorithms](https://docs.google.com/document/d/1pvmkSGk0sAp6D7Mmaq0TWuJhoJiGTR9VU53JFJTxYgo/edit?usp=sharing) (requires access)
* Practical Assignment:
  + Complete the assigned Khan material. Your instructor will check this as the Practical Assessment grade for this unit.