

(<http://csmatters.org/pd-new>) C - 06

0bC - 0b110

Advanced Algorithms



Unit Concept Lessons

Revision Date: Nov 05, 2019

Duration: 60 50-minute sessions

Lesson Summary

Summary: Teachers are introduced to the theory of computation, computability, the halting problem, and advanced algorithms. In particular, they will learn about heuristic search used by artificial intelligence (AI) programs to play games.

Outcomes:

Teachers will:

- Learn algorithms and review lesson session for lesson 5-5.
- Locate the topics about algorithms students must learn to answer each question on the practice test.
- Discuss why the CS Matters Curriculum goes beyond these requirements and what impact this has on the use of this lesson.
- define computation and some basic ideas of the theory of computation
- discuss computability and understand there are some things computers cannot solve
- explain the Halting Problem
- identify some advanced search algorithms
- understand how AI programs represent games with game trees
- understand how AI programs use uninformed and heuristic search algorithms to play games

Overview:

1. Review Lesson 5-5.
2. Computation and Algorithms
3. College Board Audit

Learning Objectives

Teacher Resources

CL06_Advanced Algorithms Folder (<https://drive.google.com/open?id=0B5vAY-fhOT-iSU82UGw5VklITzg>)

Lesson Plan

Review Lesson 5-5.

- Review Big Ideas and Computational Practices.
- Reflect on Questions by collecting questions on post-it notes for review later possibly during the TBA sessions:
 - 1 Python Programming
 - 2 Any idea or practice
- Use the College Board's curriculum framework and the spreadsheet in the safe to get access to the questions numbers from the practice test that pertains to the Learning objectives in the lesson. Distribute the

identified problems to pairs of teachers. Teachers are to assess what topic about algorithms students must possess to answer each question.

- Read the objectives of the CS Matters lesson.
- Discuss the importance of understanding algorithms beyond the minimum established by the practice test provided by the College Board. Goal is to keep a balance. These topics are potentially very important and yet could stress some students.

Computation and Algorithms

- Have teachers work in pairs. Select either a 1 or 2. Solve the assigned problem by hand. These are inverse operations yet one is much harder than the other. Computer rely on inverse algorithms to make encryption easy without making decryption easy.
- Note that the definition of computation does not depend on a computer. Instead the computer is a type or model of a system that performs computation. The first program to play the game of chess was written and tested by Alan Turing years before the construction of an electronic computer capable of using the program to play the game.
- Teachers read the CS4FUN article to find two problems that are impossible for computers to do.
- Contrast impossible problems with difficult ones. Watch the video from the Computer Science Field Guide to illustrate a type of hard or intractable problem - and maybe win a prize.
- Summarize with this video. Students are to understand both the existence of impossible and of hard problems. That impossible problems are larger than computers. Hard problems may succumb to them depending on both the quality of algorithms used and future computing advances.
- The Halting Problem video, also used by the curriculum tries to explain the halting problem. As teachers watch the video they should think of ways to help students make use of the video. Remember that balance of difficulty - the College Board will not ask students to determine if a problem is impossible for a computer to solve.
- Explain why the order DECABF is correct. D assumes the data is already sorted. E requires at most one inspection of each piece of data. C uses a halving and merging algorithm to sort smaller and smaller subgroups merging subgroups with greater and greater efficiency. The bubble sort take no advantage of splitting so it must traverse the data again and again since only one data item is guaranteed to be located properly on each pass. Factoring large integers is one of the hard

problems - provide the original number is large enough. Hard is better than impossible so F is last.

- Big O is beyond the course and should be used in a general way only as a way of quantifying the previous discussion. Log n asks evaluates to the exponent or power of its base needed to reach the sample size. The binary search needs to half the list until only one value is left. This is the same number of times as the power of two needed to reach the sample size. Big O **is not** part of the AP CSP course as defined by the College Board.
- In session two, we focus on one type of problem - developing a program smart enough to win common games. Not all of these require heuristics but some do.
- Count off from 1 to 9 and have teacher draw the next step - game trees showing all possible second moves. Go back to the linked game tree to see the total number of trees needed to represent every possible game. This game can be easily and fully described so a computer can find the optimum strategy for every move and never lose or fail to win when given an opportunity. It may seem a little hard but to a computer, the algorithm that solves tic tac toe is very simple.
- Consider using the calculators provided in the Computer Science field guide for students to see the magnitude of these numbers. When they were little they may have been taught the magic word was please but as far as taking on hard problems with algorithms the magic word is heuristic.
- Other algorithms use AI and heuristics to play a variety of games There are a few common algorithms we ask students to learn about. The mechanics of each algorithm is beyond the lesson rather their relative reliance on heuristics.

College Board Audit

- Review the College Board Audit process.
- <https://apcentral.collegeboard.org/courses/ap-computer-science-principles/course-audit> (<https://apcentral.collegeboard.org/courses/ap-computer-science-principles/course-audit>)
 - Getting to Know the Course and Exam
 - Endorsed Providers
 - Summary of Curricular and Resource Requirements
 - Curricular Requirements

- Resource Requirements



(<http://www.umbc.edu/>)



(<http://www.umd.edu/>)



(<http://www.nsf.gov/>)

Authored by: CS Matters in Maryland

Website: csmatters.org (<http://csmatters.org>)

Email: csmattersinmaryland@gmail.com (<mailto:csmattersinmaryland@gmail.com>)

This work is licensed under a
Creative Commons Attribution-ShareAlike 3.0 United States License
(<http://creativecommons.org/licenses/by-sa/3.0/us/>)
by University of Maryland, Baltimore County (<http://umbc.edu>) and University of
Maryland, College Park (<http://umd.edu>).