Accredited

AS and A LEVEL

Topic Exploration Pack

H046/H446

COMPUTER SCIENCE

Theme: Algorithms

August 2015

OCR
Oxford Cambridge and RSA

We will inform centres about any changes to the specification. We will also publish changes on our website. The latest version of our specification will always be the one on our website (www.ocr.org.uk) and this may differ from printed versions.

Oxford Cambridge and RSA Examinations is a Company Limited by Guarantee.
Registered in England. Registered company number 3484466.

Registered office: 1 Hills Road
Cambridge
CB1 2EU

OCR is an exempt charity.

Contents

This Topic Exploration Pack should accompany the OCR resource ‘Algorithms’ learner activities, which you can download from the OCR website.

Introduction

This topic introduces the concept of algorithms and looks at a range of activities that allow learners to understand a subject which can be difficult. All activities are setup with the understanding that this is an introduction to the different concepts covered within this topic, but any prior knowledge and or equipment will be clearly outlined.

Activity 1

Many learners are being taught what an algorithm is at a very early stage within computing theory, but many learners still struggle with the main concept of what an algorithm is, what makes a good algorithm and also how to develop algorithms that have been designed to improve on the concept of complexity. Activity 1 gets learners to sort objects and document their instructions in turn working towards making a basic sorting algorithm.

Resources needed:

- Cards
- Number cards
- Random classroom items (rubbers, rulers etc)
- Different colour and sized straws
- Cups
- Any item that can be sorted based on colour and size.

Task 1

Place on each learners desk a selection of objects that they will be required to sort. Do not inform the learners how to sort the objects just allow learners to sort the items in any manner they please. On completion of the first part of this task learners are required to buddy up with each other and discuss and sort out the combined objects.

Task 2

As a class, conduct discussions into the methods used to conduct the task of sorting the different objects provided to learners. The following key questions should be covered to extract deeper learning:

- Do you know of another instance where objects might be sorted like this?
- How would this be different or similar for a computer?

- What would happen if you had to sort the objects by a different aspect to yours?
- How would that have an impact?
- If you were to tell someone about how to complete your sorting, what would they have to do?

During the discussion the key aim is to discuss that the learners have used a set of instructions to solve the problem of sorting, and that they completed the task using an algorithm of some sort.

Task 3

Choosing one of the provided learner sheets (Learner Sheet A, B or C), learners are required to produce their sorting algorithms using either flowcharting or pseudocode depending on what they feel most comfortable with. Learners will be given the opportunity to share their ideas and instructions so all learners will be exposed to both methods of planning.

Task 4

Learners should be put into small groups. The learners should discuss their solutions and also research any solutions online that may be similar to their algorithm. Many learners will discover that their solutions are very similar to the common sorting algorithm, bubble sort.

Activity 2

This activity is used to expose learners to the key algorithms they will be required to both program and understand within the course so it is very important that learners are able to use and explain the algorithms. Programmed solutions in python 3 will be provided for teachers coupled with learner sheets to read over and test out programming code. Learners will require prior knowledge of programming to be able to access the content for this activity.

Task 1

In pairs, learners are given either the programming code for bubble sort or quick sort. **Learners aren't told what sort they are given, just the code**. Using the code provided and the learner task sheets, learners are required to test out the code and explain in their own words what they feel the programming code is doing with the list.

Once learners have been provided with enough time to work on their provided sorting algorithm, learners will discuss in small groups the two different sorting algorithms, what they feel the key differences are between the two, and how they operate.

Task 2

Going back to pairs learners are provided with a range of lists to place into their sorting algorithms to document what happens within the algorithm with different types of lists.

Learners are required to complete the learner task sheet which contains key questions linked to areas of interest related to how the sorting algorithms operated, which will look at best case and worst case scenarios with the list sorting.

Task 3

Learners are informed of the names of the sorting algorithms that they were provided. As a class, a list is to be placed onto the board and class based discussion is to be conducted to aid in discussing what algorithm they would use and why.

Teacher to make up a list on the board as they see fit.

Extension: Discuss if learners can think of a totally different algorithm that could be used and why they think that would be stronger.

Activity 3

Algorithm design within computing is very important based on how well the algorithm is set up and programmed. Many factors such as memory space used and how long it takes to complete a given task are affected by this. The notion of complexity is an important concept that needs be understood in relation to O notation.

https://justin.abrah.ms/computer-science/big-o-notation-explained.html

Task 1

The following clip provides an interesting introduction to the concept of complexity. Using the learner task sheet provided and stopping at key points for Q/A with the class will provide a good introduction to supporting deeper learning within this subject matter: **Watch the first 5 minutes of the video.**

http://www.youtube.com/watch?v=V6mKVRU1evU

Task 2

Using the points discussed within the first part of the task, learners are given a task sheet with increasingly more difficult problems to solve. Learners are required to look at programming code, to design on the O notation that would be used and point out areas of interest on code that justify their answers.

Task 3

As a way of developing the understanding of this concept, place learners into small groups. Learners are required to think of real life problems that they would be able to label with a big O notation. (Teacher may need to provide more than one example to aid in the understanding of the task.)

Example:

O(n): reading a book where n is the number of pages. It is the minimum amount of time it takes to read a book.

Activity 4

This activity looks at different data structures and how they will be used within real life examples where there is more of a focus on programming. The two main data structures being covered within this activity are stacks and queues. Learners will be provided with code to test out and use and will have to implement their own stack and queue for different real life scenarios

Task 1

To aid in supporting with this task, teacher and learner code will be provided. Teacher code contains detailed commenting to allow for better understanding to provide more in depth support to learners as they complete the task.

Learners will have code to test out and learner sheets to complete for this part of the activity. Using the code provided, learners will learn the basics linked to stack and queue data structures and how they are programmed to aid the completion of upcoming tasks within the activity.

Task 2

Working in pairs, learners will choose to program one of the following problems:

- Game login
- Call centre
- Robot plate clearer at restaurant
- Robot librarian putting books away.

Learners will be programming using the concept of “pair programming”. Working in pairs, learners will take turns programming. One learner will program for five minutes while the other learner supports and notes down all the errors made by the programmer. After time has passed, learners swap roles and continue on with the programming of the solution. This activity is designed to develop stronger programming skills.

OCR customer contact centre
General qualifications
Telephone 01223 553998
Facsimile 01223 552627
Email general.qualifications@ocr.org.uk