OCR Computer Science A Level

2.2.2 Computational Methods

Concise Notes

Specification:

2.2.2 a)

- **Features that make a problem solvable by computational methods**

2.2.2 b)

- **Problem recognition**

2.2.2 c)

- **Problem decomposition**

2.2.2 d)

- **Use of divide and conquer**

2.2.2 e)

- **Use of abstraction**

2.2.2 f)

- **Solving problems using:**
  - Backtracking
  - Data mining
  - Heuristics
  - Performance modelling
  - Pipelining
  - Visualisation

Features that make a problem solvable by computational methods

- Identifying if a problem can be solved using computational methods is the first stage of problem solving
- A computable problem can be solved using an algorithm
- Some computable problems are impractical to solve due to the resources (processing power, speed and memory) or time they need

Problem recognition

- Stakeholders state what they require from the solution
- Information used to clearly define the problem and system requirements
- Problem may be defined by considering:
    - Strengths and weaknesses of current solution
    - Inputs, outputs, stored data and volume of data

Problem decomposition

- Problems are broken down into smaller problems until each subproblem can be represented as a self-contained subroutine
- Decomposition reduces problem complexity by splitting it up into smaller sections
- Enables programmers to find sections that can be implemented using pre-coded modules or libraries
- Makes the project easier to manage
- Teams can be assigned different modules depending on specialisms
- Modules can be designed, developed and tested individually before being combined
- Makes it possible to develop modules in parallel, therefore deliver projects faster
- Simplifies debugging process, as it is quicker to identify, locate and mitigate errors

Use of divide and conquer

- Problem-solving technique that can be broken down into three parts:
    - Divide
        Halves the size of the problem with every iteration
    - Conquer
        Each subproblem is solved, often recursively
    - Merge
        Solutions to the subproblems are then recombined
- Applied to problem-solving in quick sort, merge sort and binary search
- Simplifies complex problems very quickly

Use of abstraction

- Excessive details are removed to simplify a problem
- Problems may be reduced to form problems that have already been solved
- This allows pre-programmed modules and libraries to be used
- Levels of abstraction allow a complex project to be divided into simpler parts
- Levels can be assigned to different teams and details about other layers hidden
- Makes projects more manageable.
- Abstraction by generalisation used to group together sections with similar functionality
- Means segments can be coded together, saving time
- Abstraction is used to represent real-world entities with computational elements

Problem solving strategies

Backtracking

- Problem-solving technique implemented using algorithms, often recursively
- Methodically builds a solution based on visited paths found to be correct
- If a path is found to be invalid, algorithm backtracks to the previous stage

Data mining

- Technique used to identify patterns or outliers in large sets of data collected from a variety of sources, termed big data
- Spots trends or correlations between data which are not immediately obvious
- Insights from data mining can aid predictions about the future
- This makes data mining a useful tool in assisting business and marketing decisions

Heuristics

- Non-optimal, 'rule-of-thumb' approach to problem-solving
- Used to find an approximate solution when the standard solution takes too long to find
- Solution found through using heuristics is not perfectly accurate or complete
- Used to provide estimations for intractable problems, shortest path-finding problems, machine learning and language recognition

Performance modelling

- Mathematical methods used to test various loads on different operating systems
- Provides cheaper, less time-consuming or safer method of testing applications
- Useful for safety-critical computer systems, where a trial run is not feasible

Pipelining

- Process in which modules are divided into individual tasks, with different tasks being developed in parallel
- Enables faster project delivery
- Output of one process typically becomes the input of another, resembling a production line
- Commonly used in RISC processors: different sections of the Fetch-Decode-Execute cycle are performed simultaneously

Visualisation

- Data can be presented in a way that is easier for us to understand using graphs, trees, charts etc.
- Makes it possible to identify trends that were not otherwise obvious