2.2.2 Computational Methods

Revision sheet

1 Problems

Not all problems can be solved by a computer. The first stage of problem solving is identifying whether or not a problem can be solved using computational methods. Problems which can be solved using a computer are called 'computable', they must be able to be solved within a finite and realistic amount of time; typically they will consist of inputs, outputs and calculations. Some problems, although technically 'computable' may be impractical to solve in this way.

1.1 Problem Recognition

After a problem has been deemed 'computable', the next step is to clearly identify what the problem is. Stakeholders state what they require from the finished product, this information is used to clearly identify the system requirements. Requirements may be defined by: analysing the strengths and weaknesses of current solutions; or considering types of data involved including inputs, outputs, stored data and amount of data.

1.2 Problem Decomposition

Once a problem has been identified, it is continually broken down into smaller problems. This continues until each sub-problem can be represented as a self-contained subroutine. By decomposing the problem like this, programmers may find that certain sections of the program can be implemented using pre-coded modules or libraries which will save time and money. Decomposition also makes the problem easier to manage, as different, specialised software development teams can tackle different aspects to the solution which they specialise in. The use of subroutines here means that each section can be designed and tested on its own before being combined together, making testing and debugging easier.

2 Divide And Conquer

This is a problem solving technique which can be divided into three parts: divide (halving the size of the problem with every iteration); conquer (solve these problems); and merge (recombine the solutions to the problem to give the overall solution). The biggest advantage of using this approach to problem solving is that with every iteration, the problem is halved which simplifies a complex problem.

3 Use Of Abstraction

Representational abstraction is used when solving problems, this is where excessive details are removed to simplify a problem. Abstraction allows programmers to focus on the core aspects required of the solution, rather than worrying about the unnecessary details. Using multiple levels of abstraction allows a large complex project, along with its functionality, to be split up into simpler component parts. These component parts can then be dealt with by different teams - making projects more manageable. Abstraction by generalisation may also be used to group together different sections of the problem with similar underlying functionality, allowing for segments to be coded together and reused - saving time.

4 Problem Solving Strategies

There are a number of different problem solving strategies.

4.1 Backtracking

This is often implemented recursively in algorithms. It works by methodically visiting each path and building a solution based on the paths found to be correct. If a path is found to be invalid at any point, the algorithm backtracks to the previous stage and visits an alternate path.

4.2 Data Mining

This is a technique used to identify patterns or outliers in large data sets, termed big data (which is usually collected from a variety of sources). Data mining is used in software designed to spot trends or identify correlations between data which are not immediately obvious. The insights provided from data mining can be used in many ways. One such use is to inform shops about peoples shopping habits, allowing supermarkets to release offers which may tempt a customer to purchase something which they did a lot historically but have not done so recently. Data mining involves the handling of personal data, it is crucial that it is dealt with the present legislation (Data Protection Act and GDPR).

4.3 Heuristics

These are non-optimal, rule of thumb approaches to problem solving which are used to find an approximate solution to the problem when the standard solution is unreasonably time-consuming or resource-intensive to find. The heuristics solution is not perfect, however it is good enough. They are used for providing an estimate solution for intractable problems - problems for which the solution takes an unreasonably long amount of time to be found.

4.4 Performance Modelling

This eliminates the need for true performance testing by providing mathematical methods to test a variety of loads on different operating systems. PM provides a cheaper, less time consuming or safer method of testing applications. This is extremely useful for safety critical applications where it is not safe to do a real life trial run before the system can be implemented. The results of PM can help companies judge the capabilities of a system, how it will cope in different environments and assess whether it is safe to implement.

4.5 Pipelining

This is a process that allows for projects to be delivered faster as modules are divided into individual tasks with different tasks being developed in parallel.

4.6 Visualisation

Data can be presented in a way that is easier for us to understand using visualisation. This makes it possible to identify trends that were not otherwise obvious, particularly amongst statistical data. This is another technique used by businesses to pick up on patterns which can be used to inform business decisions.