

2.2 PROGRAMMING TECHNIQUES · 2.2.2(D)–(F)

Divide & conquer, backtracking, heuristics

The computational-methods toolkit. Spec 2.2.2(d)–(f).

01 Divide & conquer

What Repeatedly split into same-type sub-problems, then **combine**.

Examples Binary search · merge sort · quicksort.

02 Abstraction & backtracking

Abstraction Remove detail → simpler model (e.g. graph).

Backtracking Try a step; **undo** dead ends, try another (maze, Sudoku).

03 Data mining & heuristics

Data mining Find hidden **patterns** in large data.

Heuristic Rule of thumb → good-enough, not guaranteed best (sat-nav route).

04 Modelling, pipelining, visualisation

Perf. model Predict behaviour with a model (too costly to test for real).

Pipelining Overlap stages of a sequence → throughput.

Visualisation Show data graphically → spot patterns.

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Six slips on computational-methods questions.

01

Divide & conquer: split same-type sub-problems + **combine** (not just decompose).

02

Heuristic = **good-enough, fast** — not guaranteed optimal.

03

Backtracking **undoes** dead ends — not blind brute force.

04

Data mining = hidden **patterns** in big data, not finding one record.

05

Pipelining **overlaps stages**; ≠ parallel processing (separate tasks).

06

Learn one **example** per method — questions ask you to spot it.