

2.3 ALGORITHMS · 2.3.1(F)

Dijkstra's algorithm & A*

Shortest path in a weighted graph. Spec 2.3.1(f).

01 Dijkstra's algorithm

Finds	Shortest path from start to all nodes.
Needs	Weighted graph (non-negative weights).
Step	Settle the nearest unvisited node; relax neighbours.
Record	Distance + previous node (to trace path).

02 A*

Finds	Shortest path to one goal , faster.
Adds	A heuristic h (estimate to goal).
Picks	Smallest $f = g + h$.
g / h	g = distance so far · h = estimate left.

03 Key facts

Relax	An indirect route can beat a direct edge ($A \rightarrow C \rightarrow B = 3 < A \rightarrow B = 4$).
A* speed	Steers towards goal → explores fewer nodes .
A* guarantee	Shortest path if h never overestimates .
Link	Dijkstra = A* with $h = 0$. A* = sat-nav / game routing.

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Six slips on shortest-path questions.

01

These need a **weighted** graph — BFS only does unweighted.

02

Always settle the **smallest current distance** unvisited node.

03

A cheaper **indirect** route can beat a direct edge — relax!

04

A* uses $f = g + h$: g = so far, h = estimate to goal.

05

A* is faster (fewer nodes) but needs a non-overestimating **h** .

06

Dijkstra = A* with $h = 0$.