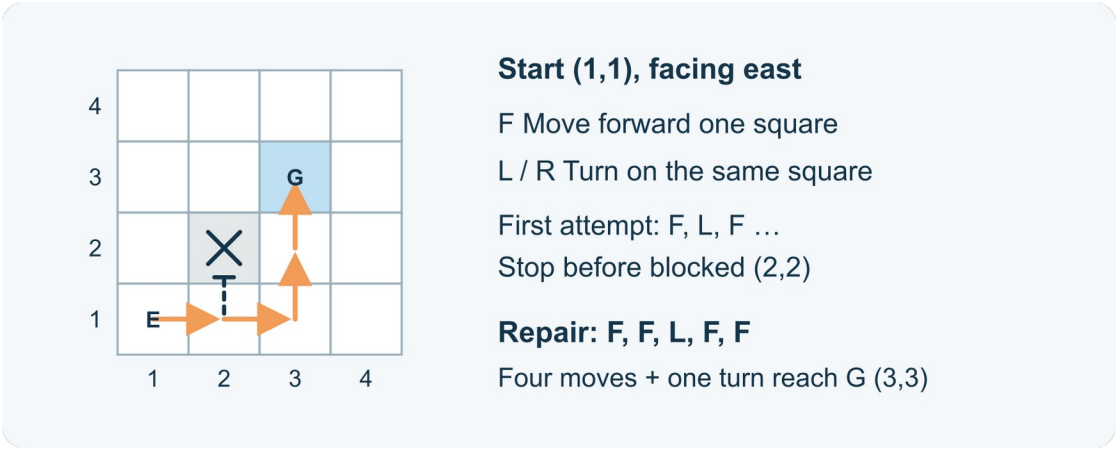
Debug a paper robot

Where does an instruction sequence first stop matching its goal?

Give a paper robot a starting direction, trace its commands and repair one route. No coding software is needed.

Investigate | 15–25 minutes, or one trace | Paper and a pen or pencil, A small paper arrow, or a drawn arrow you can redraw

Start here Draw a 4-by-4 grid. Columns run left to right and rows bottom to top, both 1 to 4. Put an east-facing arrow at (1,1), a goal at (3,3) and a block at (2,2).



A turn changes direction, not position

1 Set the command language

F means move one square in the direction the arrow faces. L means turn left 90 degrees on the same square; R means turn right 90 degrees on the same square. Turns do not move the robot. Stop before a move would enter the block or leave the grid.

2 Trace the first program

Try F, L, F, F, R, F from the stated start. Record the square and facing direction after each command. At the first blocked move, stop and mark that command instead of completing the rest.

3 Make a repair

Reset the arrow to (1,1), facing east. Try F, F, L, F, F, or design your own sequence. Record the trace again. Reaching the goal is useful information about this starting case.

4 Explain the repair's limits

Describe which instruction caused the original problem and why the replacement route avoids it. If you want another test, change the starting direction on paper and predict whether the same commands still work.

Exact example locate the first blocked move

This robot and its board are invented. East is right, north is up. The goal is (3,3); the robot begins at (1,1), facing east.

• F, L, F…: move to (2,1) east; turn north at (2,1); the next F would enter blocked (2,2), so stop before it.

• Repair F, F, L, F, F: (2,1) east → (3,1) east → (3,1) north → (3,2) north → (3,3) north.

• The repair reaches this goal in five commands: four moves and one turn.

What this example can show The repair is tested by tracing this one starting case. It is not a general route-finding algorithm, a real robot test or proof that every changed board will work.

A different way

Use the supplied trace and point to the first problem, or have someone move the arrow exactly as you direct. A smaller empty grid is another valid starting point.

If you want to go further

Write a repeat instruction for two forward moves and explain what it expands to. Test a different starting direction before claiming the shorter notation still reaches the goal.

Something to notice

Did I check the robot's direction as well as its square?

Read or explore more

Horizon topic Computing

https://launchhorizonhq.com/learn/computing/

Horizon topic Algorithms Debugging

https://launchhorizonhq.com/learn/algorithms-debugging/

Horizon topic Loops Repetition

https://launchhorizonhq.com/learn/loops-repetition/

CS Unplugged, University of Canterbury Computational Thinking and CS Unplugged

https://www.csunplugged.org/en/computational-thinking/

Computer Science Field Guide, University of Canterbury Algorithms

https://www.csfieldguide.org.nz/en/chapters/algorithms/

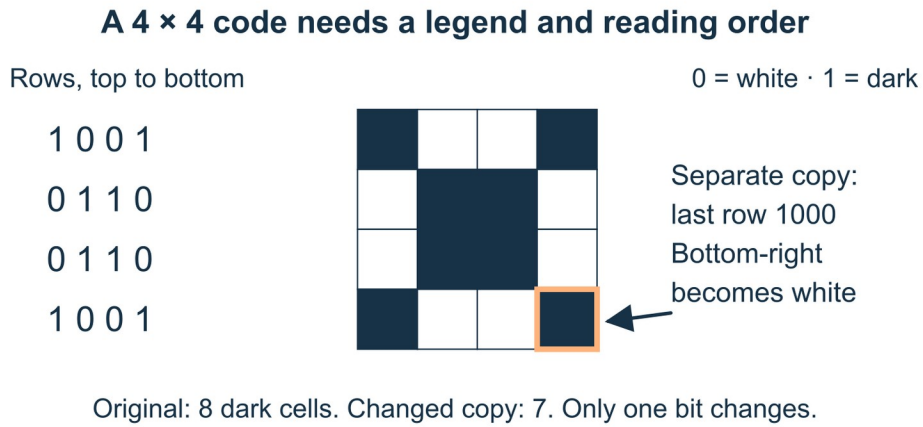
Encode a tiny pixel picture

Can someone rebuild a small image from an unambiguous written code?

Decode an original four-by-four bitmap with a complete legend, then check how one changed bit alters the picture.

Make | 15–25 minutes, or one row | The supplied example, read or heard, Optional scrap paper and a pen or pencil

Start here Use a 4-by-4 grid and these rows from top to bottom: 1001, 0110, 0110, 1001. Read each row left to right; 1 is dark and 0 is white.



A code needs an arrangement rule

1 Keep the header with the data

Write width 4 and height 4 beside the rows. Without a size and reading-order convention, a string of digits may be arranged in different ways.

2 Decode the original

Shade cells marked 1 and leave 0 white. The pattern has dark cells along both diagonals: eight dark and eight white cells.

3 Check a round trip

Read your shaded grid back into four digit rows. Compare every row with the supplied code rather than deciding only whether the picture looks familiar.

4 Change one bit

Make a separately labelled copy whose last row is 1000. The bottom-right cell becomes white; the other 15 cells stay the same. Keep the original for comparison.

Original bitmap a tiny diagonal pattern

Header: width 4, height 4; row order top to bottom; within each row left to right; 0 white, 1 dark.

• Rows: 1001/0110/0110/1001. Each row contains four bits.

• The original contains eight digits equal to 1, so it has eight dark cells.

• Changing only the last digit produces 1000 in the last row and seven dark cells overall.

What this example can show This is a monochrome encoding exercise, not every image format. A code needs its agreed dimensions and symbol meanings to be decoded correctly.

A different way

Decode only the first row or point to the changed bottom-right cell. You can explain the supplied grid without drawing or installing anything.

If you want to go further

Describe row 1001 as one dark cell, two white cells and one dark cell. Explain why counts without a starting colour would be ambiguous.

Something to notice

Did my code preserve both the picture data and the rules for arranging it?

Read or explore more

Horizon topic Pixels Images

https://launchhorizonhq.com/learn/pixels-images/

Horizon topic Binary Digital Data

https://launchhorizonhq.com/learn/binary-digital-data/

Horizon topic Formats Compression

https://launchhorizonhq.com/learn/formats-compression/

Computer Science Field Guide, University of Canterbury Images and Colours

https://www.csfieldguide.org.nz/en/chapters/data-representation/images-and-colours/

Computer Science Field Guide, University of Canterbury Run length encoding

https://www.csfieldguide.org.nz/en/chapters/coding-compression/run-length-encoding/

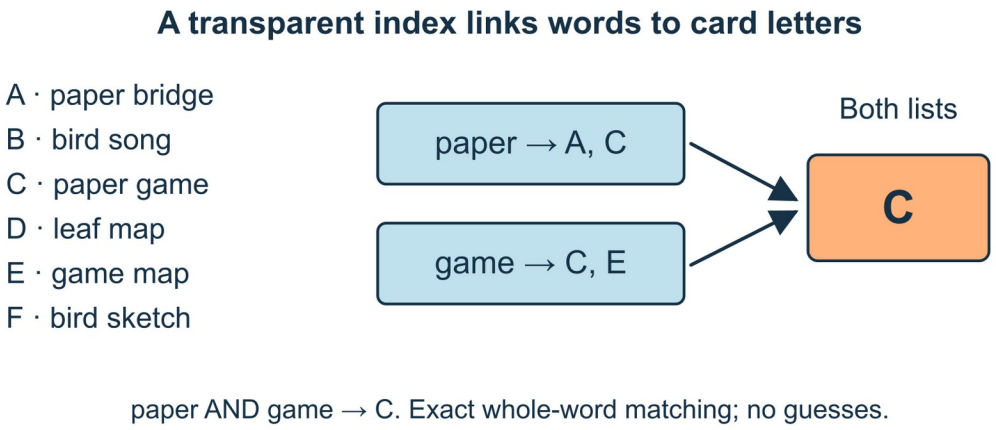
Build a six card search index

What makes a paper search return a useful match?

Build a small keyword-to-card index, combine exact matches and explain what a missing word does and does not mean.

Make | 15–25 minutes, or one query | The supplied example, read or heard, Optional scrap paper and a pen or pencil

Start here Use six invented titles: A paper bridge; B bird song; C paper game; D leaf map; E game map; F bird sketch.



Six titles, a transparent matching rule

1 Define a matching rule

Use lowercase whole words exactly as written in the titles. Index every title word; do not guess synonyms, plurals or anyone's interests.

2 Build the lists

For each word, list the matching card letters. For example paper → A, C; bird → B, F; game → C, E; map → D, E. Also include bridge → A, song → B, leaf → D, sketch → F.

3 Combine a query

For paper AND game, keep only card letters in both lists. The intersection of A, C and C, E is C. The word games has no exact match under this rule.

4 Explain the limit

A failed exact match does not show that the idea is uninteresting or unknown everywhere. It describes this small collection and this matching rule; you can edit the query.

Constructed index six complete cards

All six titles and their words are supplied. Card letters identify records, not people or user profiles.

• paper → A, C; bird → B, F; game → C, E; map → D, E.

• bridge → A; song → B; leaf → D; sketch → F.

• paper AND game returns C; games returns none under exact whole-word matching.

What this example can show This is a tiny index model, not a prediction engine or a benchmark of real search speed. It cannot search words absent from its six supplied titles.

A different way

Compare only the paper and game lists, or have the titles read aloud. No device, account, browsing or personal search history is required.

If you want to go further

Rename F from bird sketch to paper sketch. Update the bird and paper lists while keeping the sketch entry, then rerun one query.

Something to notice

Did I distinguish the collection's contents from the search rule's limits?

Read or explore more

Horizon topic Search Indexes

https://launchhorizonhq.com/learn/search-indexes/

Horizon topic Tables Sorting

https://launchhorizonhq.com/learn/tables-sorting/

Horizon topic Computing

https://launchhorizonhq.com/learn/computing/

CS Unplugged, University of Canterbury Searching algorithms

https://www.csunplugged.org/en/topics/searching-algorithms/

PostgreSQL Global Development Group Preferred Index Types for Text Search

https://www.postgresql.org/docs/current/textsearch-indexes.html