

10-13 LECTURE + INDIVIDUAL EXERCISE
13-14 LUNCH BREAK
14-14:30 INDIVIDUAL EXERCISE PRESENTATIONS
14:30-17:00 GROUP PROJECT

WHAT IS AN ALGORITHM?

**AN ALGORITHM IS A FINITE, UNAMBIGUOUS SEQUENCE OF STEPS OR
INSTRUCTIONS DESIGNED TO SOLVE A SPECIFIC PROBLEM OR
ACCOMPLISH A SPECIFIC TASK.**

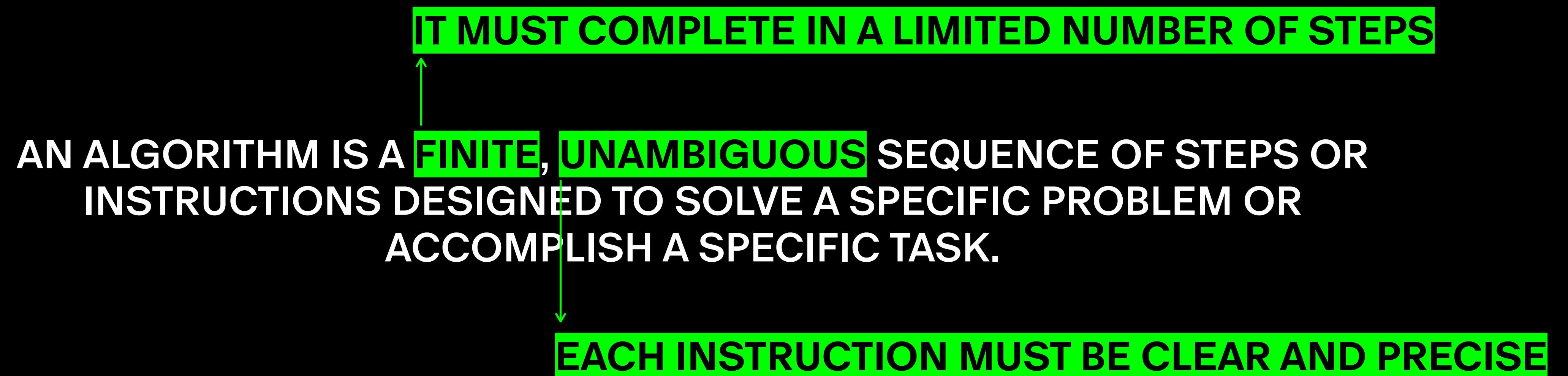
IT MUST COMPLETE IN A LIMITED NUMBER OF STEPS

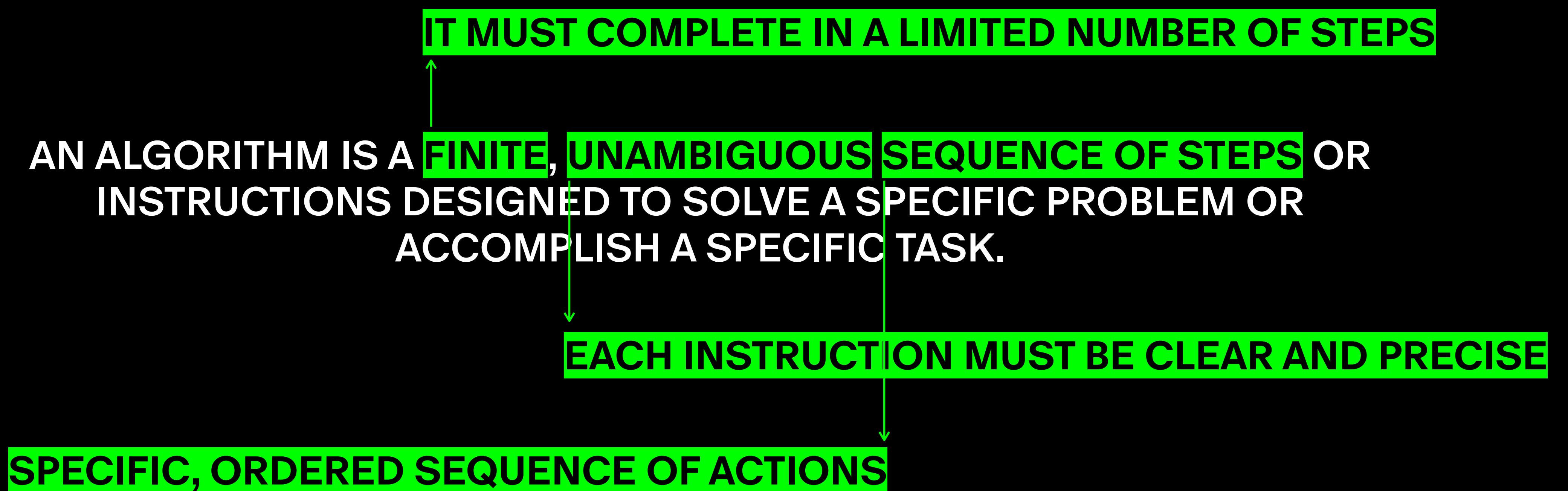
AN ALGORITHM IS A **FINITE**, UNAMBIGUOUS SEQUENCE OF STEPS OR
INSTRUCTIONS DESIGNED TO SOLVE A SPECIFIC PROBLEM OR
ACCOMPLISH A SPECIFIC TASK.

AN ALGORITHM IS A **FINITE, UNAMBIGUOUS** SEQUENCE OF STEPS OR INSTRUCTIONS DESIGNED TO SOLVE A SPECIFIC PROBLEM OR ACCOMPLISH A SPECIFIC TASK.

IT MUST COMPLETE IN A LIMITED NUMBER OF STEPS

EACH INSTRUCTION MUST BE CLEAR AND PRECISE





[0, 10, 20, 2, 5, 3, 20, 40, 50, 1, 7, 8, 9]



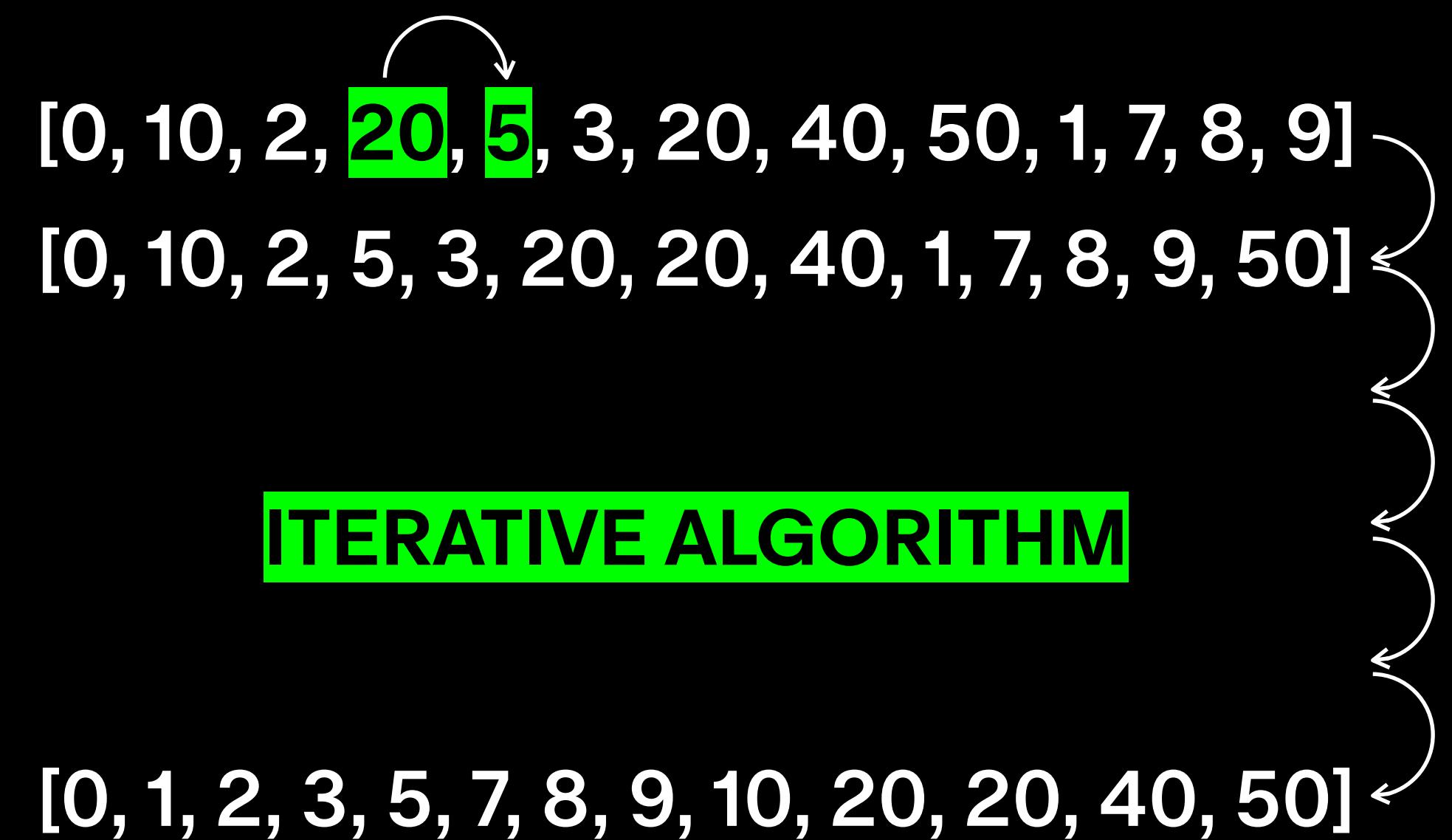
[**0**, **10**, 20, 2, 5, 3, 20, 40, 50, 1, 7, 8, 9]

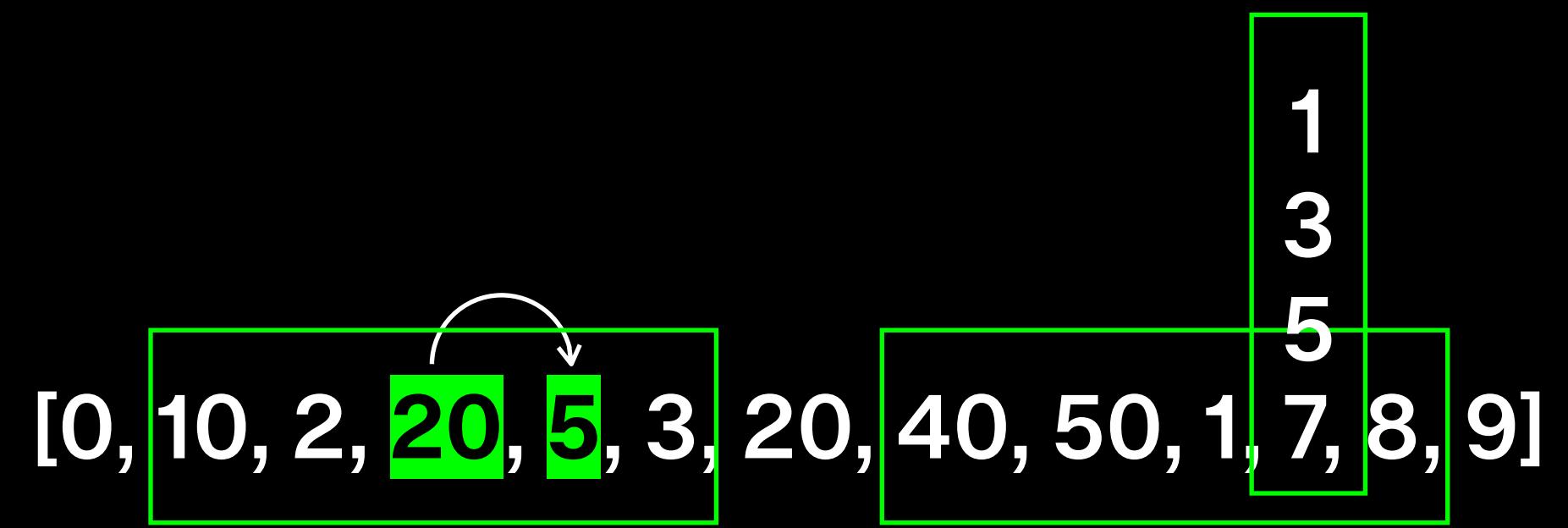
[0, **10**, **20**, 2, 5, 3, 20, 40, 50, 1, 7, 8, 9]

[0, 10, 20, 2, 5, 3, 20, 40, 50, 1, 7, 8, 9]

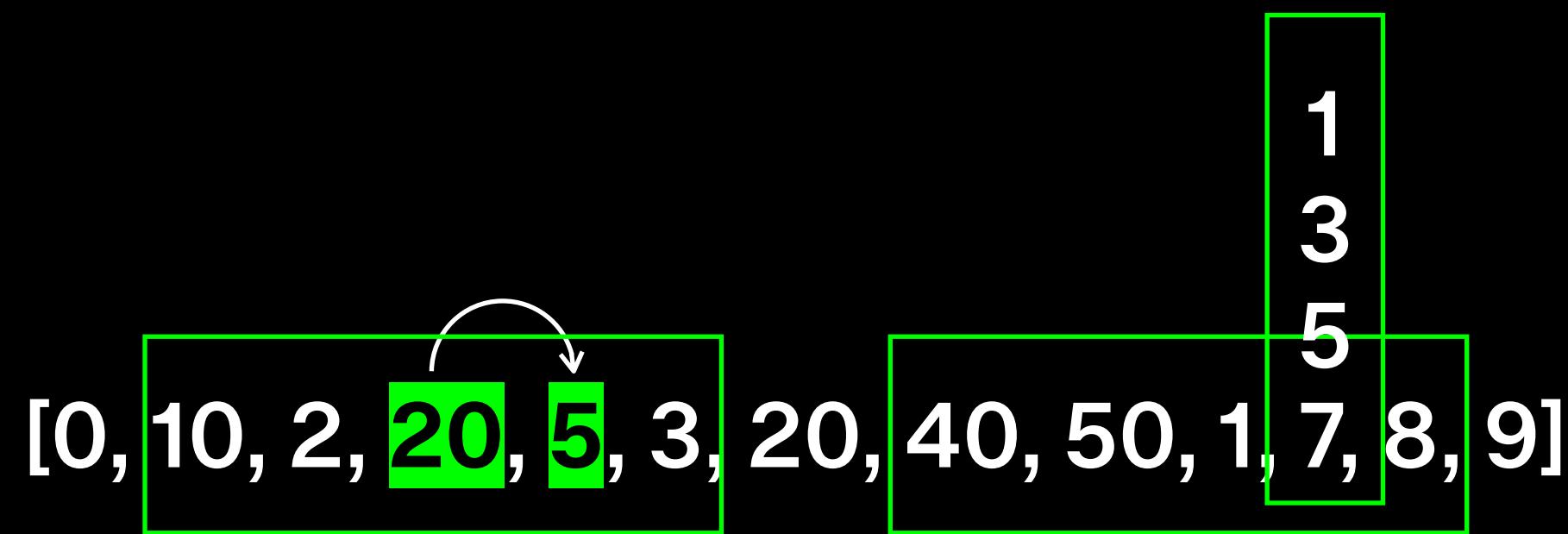
[0, 10, **2**, **20**, 5, 3, 20, 40, 50, 1, 7, 8, 9]
↑

[0, 10, 2, **20**, **5**, 3, 20, 40, 50, 1, 7, 8, 9]





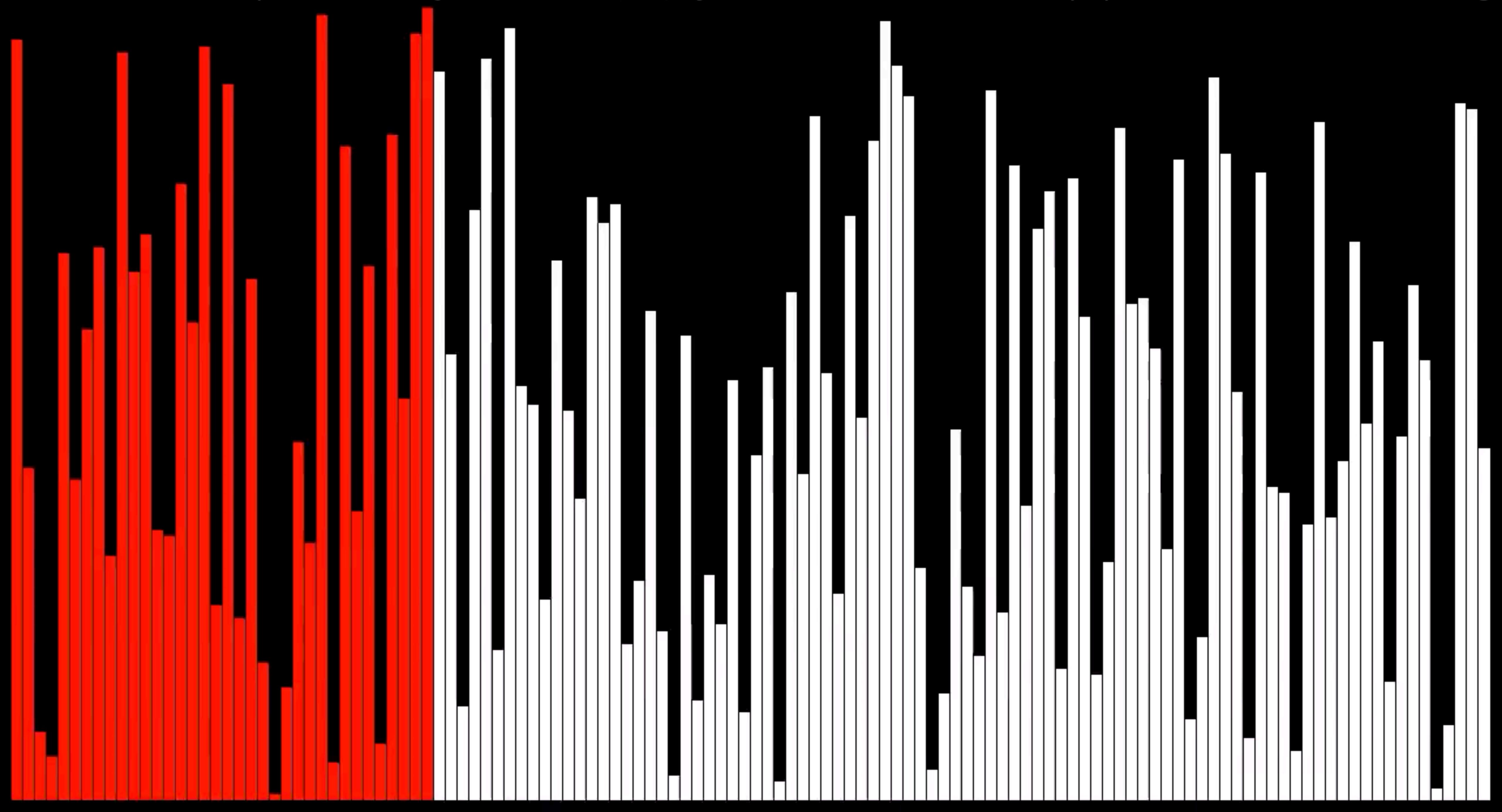
ONE PROBLEM, MANY POSSIBLE PATHS

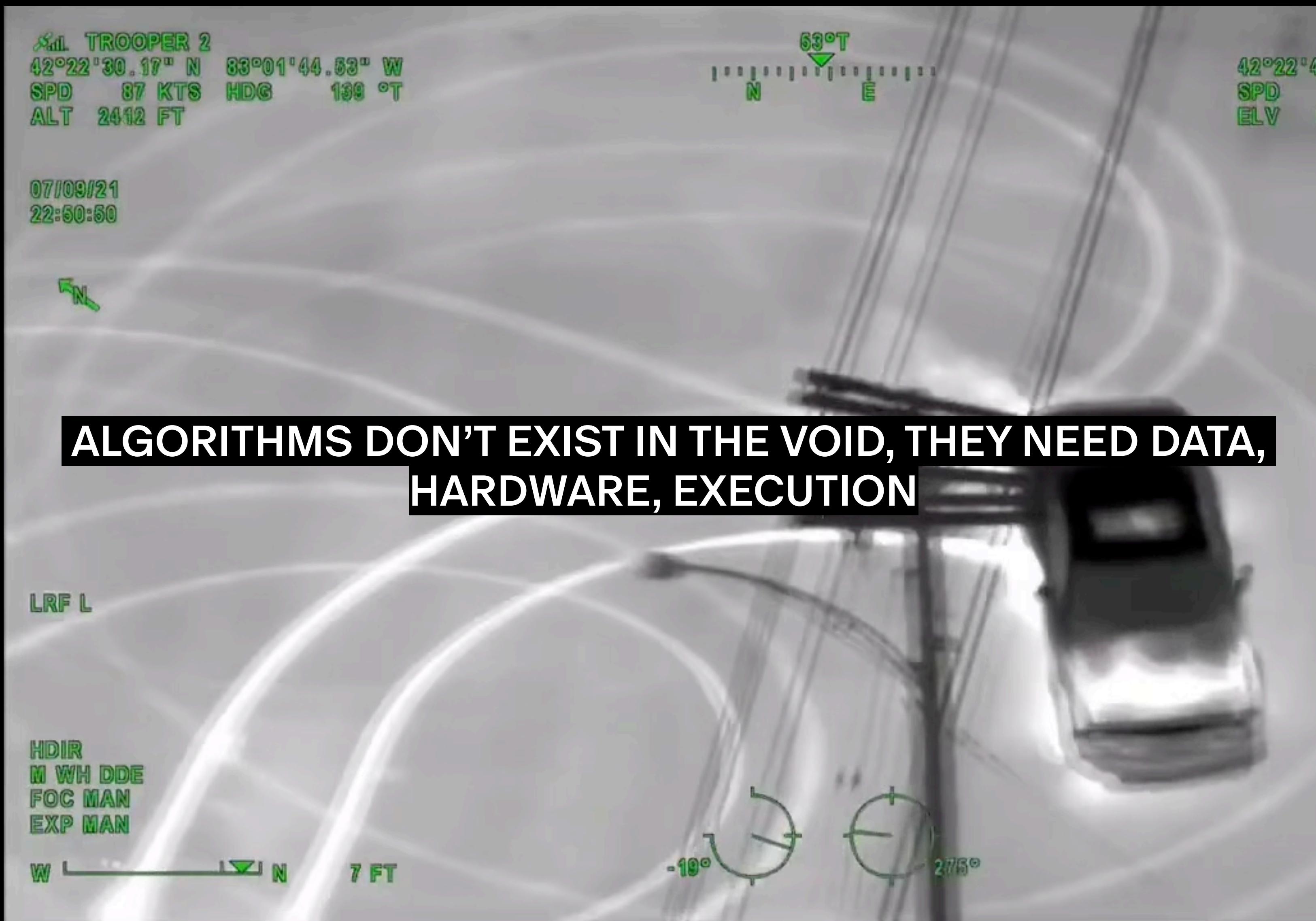


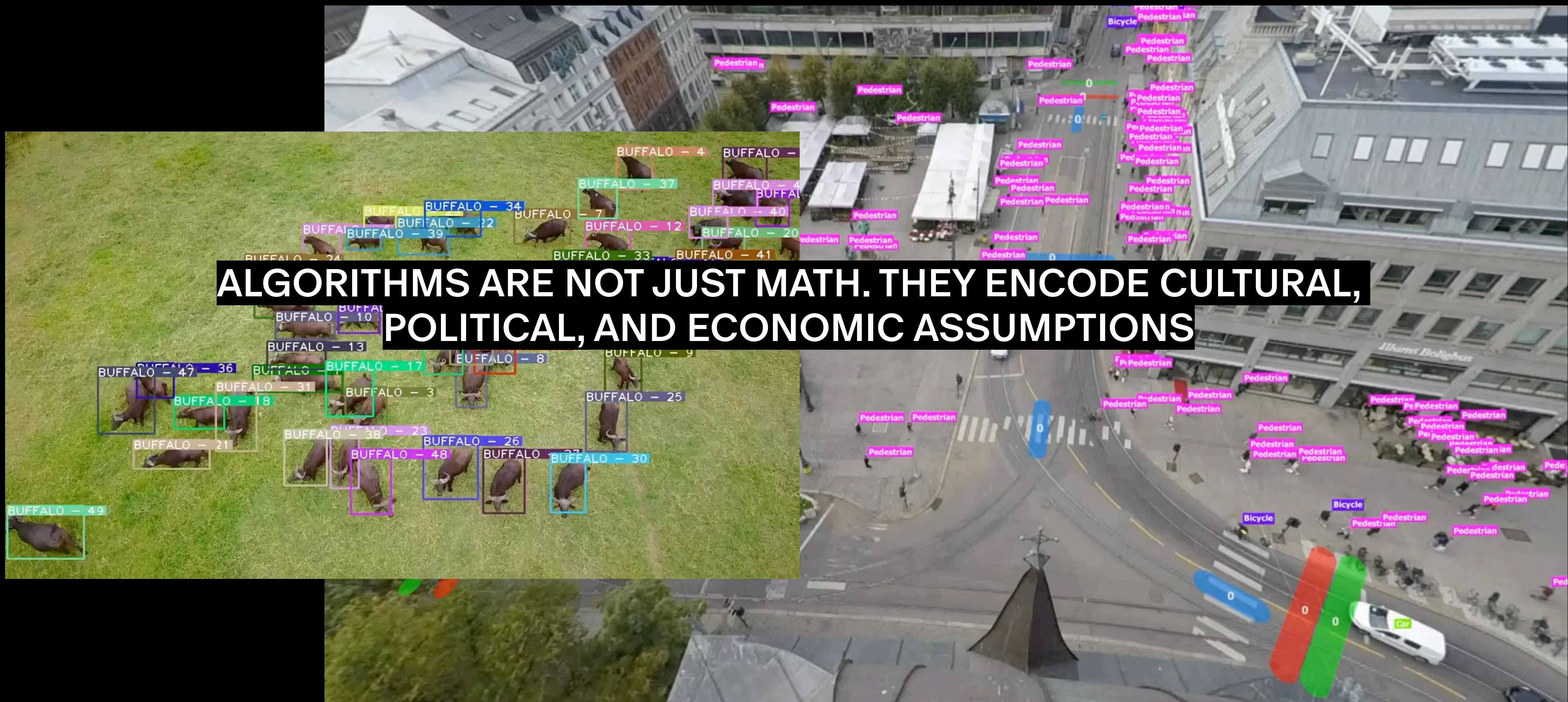
ONE PROBLEM, MANY POSSIBLE PATHS

>>TIME COMPLEXITY<<
HOW LONG THEY TAKE

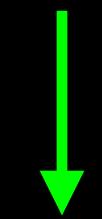
>>SPACE COMPLEXITY<<
HOW MUCH MEMORY THEY NEED







SPACE COMPLEXITY AND TIME COMPLEXITY AS GENERATIVE FORCES
AND AESTHETIC FORMS OF SYNTHESIS



WEBSpace AS ALGORITHMIC SYNTHESIS

HTML, CSS AND JS
DEV CONSOLE
PROJECTS

INDIVIDUAL EXERCISE**[1] DOWNLOAD VS CODE****[2] FORK THE GITHUB REPO <https://github.com/electro-cute-angels/digital-native/tree/main>****[3] USE THE PROVIDED HTML/CSS/Javascript STARTER TO CRAFT A WEBSITE WHERE SPACE AND TIME BECOME CREATIVE MATERIAL. YOU MUST UNDERSTAND THE CODE, TWEAK IT, REWRITE PARTS, AND AUTHOR YOUR OWN LOGIC**

WRITING EXPLICIT INSTRUCTIONS IN A FORMAL LANGUAGE (JS, PYTHON, C, JAVA,..)

CODING

A LOOSER, EXPLORATORY WAY OF CODING

~~CODING~~
VIBE CODING

GIVING NATURAL-LANGUAGE (OR MULTIMODAL) INSTRUCTIONS TO LLMS

~~CODING~~
~~VIBE CODING~~
PROMPT ENGINEERING

GOOD PROMPTS FOR CODE = SPEC + CONTEXT + CONSTRAINTS + TESTS

HELLO! PLEASE CREATE A PROJECT USING HTML, CSS, AND JAVASCRIPT.

GOAL:

- USE THE MET MUSEUM PUBLIC API.
- ALLOW THE USER TO ENTER A SEARCH KEYWORD IN A TEXT BOX.
- QUERY THE API WITH THAT KEYWORD:
[HTTPS://COLLECTIONAPI.METMUSEUM.ORG/PUBLIC/COLLECTION/V1/SEARCH?Q=KEYWORD](https://collectionapi.metmuseum.org/public/collection/v1/search?q=KEYWORD)
- FROM THE RETURNED IDS, FETCH ARTWORK DETAILS USING:
[HTTPS://COLLECTIONAPI.METMUSEUM.ORG/PUBLIC/COLLECTION/V1/OBJECTS/OBJECT_ID](https://collectionapi.metmuseum.org/public/collection/v1/objects/object_id)

BEHAVIOR:

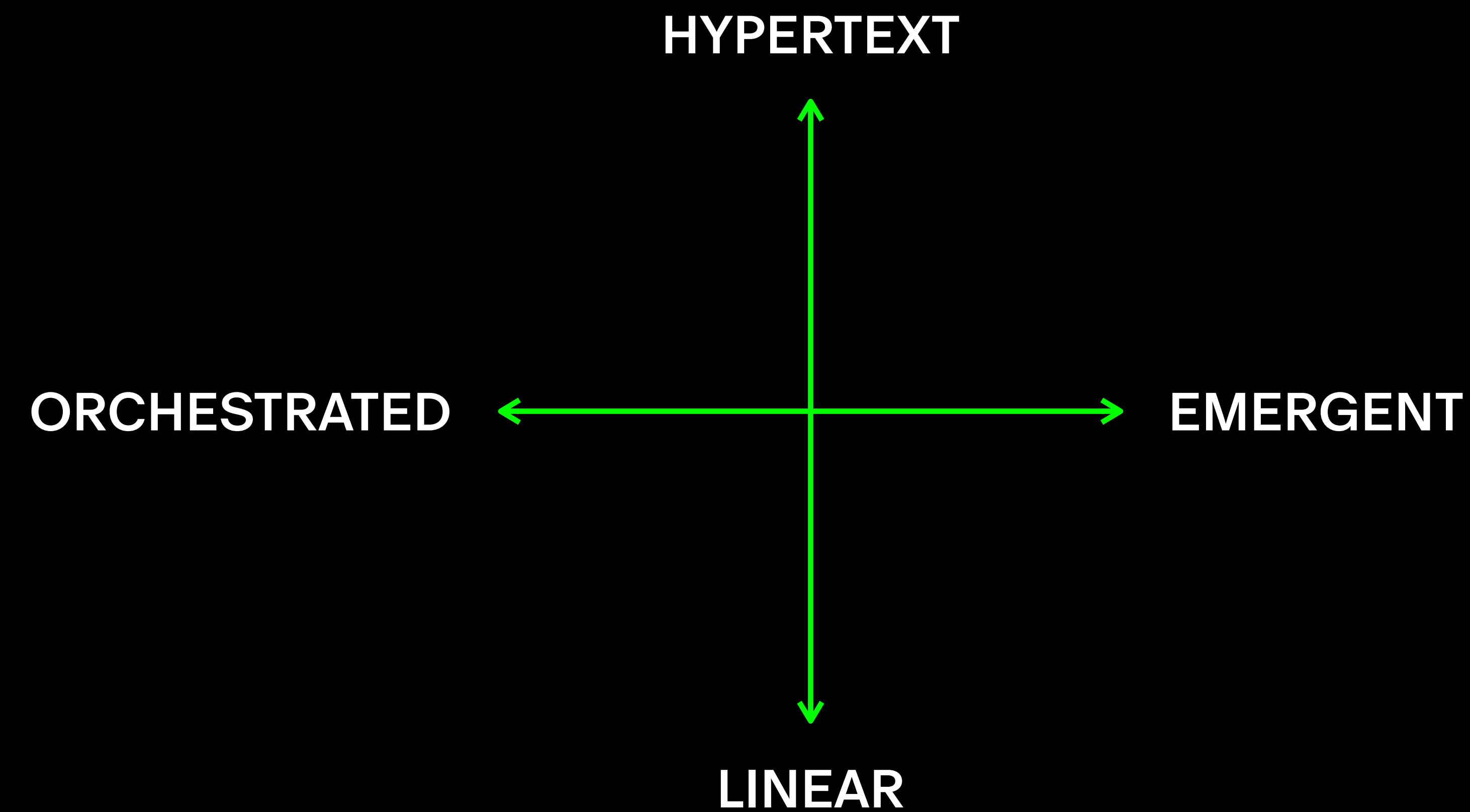
- EVERY 1 SECOND, FETCH AND DISPLAY A NEW ARTWORK IMAGE RELATED TO THE KEYWORD.
- EACH IMAGE SHOULD APPEAR (SPAWN) AT THE CURRENT CURSOR POSITION IF THE CURSOR HAS MOVED, OR AT A RANDOM POSITION ON THE SCREEN IF THE CURSOR IS IDLE.
- EACH TIME AN ARTWORK IS SPAWNED, ALSO PLAY A SHORT BEEP SOUND USING THE WEB AUDIO API.
- CONTINUE CYCLING THROUGH DIFFERENT ARTWORKS FROM THE SEARCH RESULTS, NOT ALWAYS THE SAME ONE.

AESTHETIC REQUIREMENTS:

- MINIMAL, "BACKEND-Y" VIBE.
- DARK BACKGROUND (#0A0A0A) WITH LIGHT GRAY TEXT (#EAEAEA).
- USE A MONOSPACED FONT (E.G., `MENLO, MONOSPACE`).
- SPAWNED IMAGES SHOULD BE SMALL THUMBNAILS (MAX 200PX) WITH THIN BORDERS (1PX SOLID #333).
- TEXT INPUT BOX STYLED SIMPLY (THIN BORDER, NO ROUNDED CORNERS, WHITE TEXT ON DARK BACKGROUND).
- SUBTLE HOVER EFFECT ON IMAGES (SLIGHT OPACITY CHANGE).
- FULL SCREEN

REQUIREMENTS:

- ORGANIZE THE PROJECT IN **SEPARATE FILES**:
 - `INDEX.HTML` FOR THE PAGE STRUCTURE
 - `STYLE.CSS` FOR STYLING
 - `SCRIPT.JS` FOR JAVASCRIPT LOGIC
- KEEP CODE CLEAN AND WELL-COMMENTED SO IT'S EASY FOR STUDENTS TO FOLLOW.
- INCLUDE BASIC ERROR HANDLING (IF NO IMAGE AVAILABLE, SKIP TO THE NEXT ARTWORK).



INTEGRATE GITHUB COPILOT IN VS CODE

AS PROMPT ENGINEERS
DESIGN YOUR OWN SYNTHESIS AND BUILD YOUR OWN WEBSITE