

Summary: Plotter Simulator Project

GitHub [repository](#).

[Project Structure](#).

Algorithms implemented:

- Brute Force
- Greedy Algorithm with Grid Search (good for large N)
- KD Tree Nearest Neighbor (mentioned briefly during the interview)
- R Tree Nearest Neighbor (mentioned briefly during the interview)

See [algorithm.py](#) for implementations.

Simulator modes:

- Direct – Pen moves in an arbitrary direction with a set speed.
- Independent – X & Y arms move independently, each with an identical speed limit. For motion in non-45-degree directions, one arm needs to slow down to accommodate the overall direction.

Scenarios simulated:

- Random segments
- Random segments with end points near the edges of the canvas
- A smiley face

Simulation [animations](#) uploaded to my [Dropbox](#) for easier viewing, also available on [GitHub](#).

- The filenames denote the scenario simulated and the simulator mode.
- Helped me identify a significant bug in implementing the nearest neighbor search.

Visualizations of plotter instructions from algorithms uploaded to [GitHub](#). Navigate to [plotter_simulation/output/](#) for more.

Performance comparisons (see next page):

- Based on the “random segments” scenario.
- Efficiency becomes higher when endpoints are only near the margins.

Algorithm Path Comparison

