5. Use Git Bisect for Efficient Bug Hunting in Version Control

Why: Manually checking commits for bugs is time-consuming. Git bisect automates binary search on commit history, pinpointing the exact commit introducing a bug, which accelerates debugging in collaborative projects and ensures faster releases.

How to Implement: Run `git bisect start`, mark good/bad commits with `git bisect good/bad`, and test until it identifies the culprit. Integrate with automated tests for non-interactive mode. Use in repos with linear history for best results.

```bash
# Hack: Git bisect workflow
git bisect start  # Start bisect mode
git bisect bad    # Current HEAD is bad
git bisect good v1.0  # Known good commit/tag
# Git checks out midpoint; test your code
# If bug present: git bisect bad
# If not: git bisect good
# Repeat until bisect identifies the bad commit
git bisect reset  # Exit bisect mode
```

Analysis: For a repo with 1000 commits, bisect finds the issue in ~10 steps (log2 scale) vs. linear search's 500+. It's non-destructive and works with branches, making it a staple in CI/CD pipelines for maintaining code quality.