🌟Nudamu AGI Prototype — Project Plan & Snapshot

🦜Current Snapshot (What We Have)

- **Benchmark Suite:** Mixed math + ARC grid test cases.
- **Rule Formats:** Can express math identities, grid transformations, and hybrid rules.
- **CognitiveReasoner Plan:**
- Rule induction (induce_rules)
- Program synthesis (synthesize_programs)
- Meta-reasoning loop (meta_reasoning_loop)
- Failure diagnosis + rule adjustment (diagnose_failure, adjust_rules)
- Knowledge base management (long_term_memory, working_memory)
- **Placeholder methods:** Generalize deltas, apply transformations, execute programs.

🌳Planned Update (What We Will Build)

🦠Phase 1: Basic Engine

- Build `CognitiveReasoner` skeleton.
- Implement `solve()` to induce mock rules + return simple solutions.
- Add `execute()` to apply basic transformations (math string eval / grid mock rule).

🦠Phase 2: Real Rule Induction

- Implement `find_delta` for grids + math structures.
- Generalize deltas into variable-based rules.
- Apply rules to produce outputs.

🦠Phase 3: Meta Loop + Failure Diagnosis

- Implement meta-reasoning loop: generate, test, score candidates.
- Diagnose failures, adjust or refine rules.
- Track best candidate solution.

🦠Phase 4: Benchmark + Parallelism

- Build benchmark runner for full suite.
- Enable parallel execution for program search.
- Collect accuracy, speed metrics.

🦏Backup Notes

- All code is under clean structure: `core/`, `data/`, `utils/`, `tests/`.
- Current files include reasoning engine, rule mapper, pattern detector, dataset loader, submission writer.
- Ready to extend with meta-reasoner, program synthesizer, rule inductor modules.

Next Action

👉Ready to kick off **Phase 1**? I can draft `CognitiveReasoner` starter code for you to save + run.