Minishell AST (Abstract Syntax Tree) Checklist

This checklist covers the implementation of the Abstract Syntax Tree (AST) used to structure and execute shell command logic in Minishell.

- [ ] Define t_command_tree with left/right and type
- [ ] Support node types: N_EXEC, N_PIPE, N_ANDIF, N_OR
- [ ] Create AST node creation functions (e.g., create_exec_node())
- [ ] Assign left/right children correctly during parsing
- [ ] Wrap simple commands into N_EXEC with t_exec as data
- [ ] Build tree recursively from token list
- [ ] Respect operator precedence: ( ), &&, ||, |
- [ ] Handle grouping with parentheses as subtrees
- [ ] Ensure AST structure is testable with example inputs
- [ ] Implement execution logic that traverses AST recursively
- [ ] Free entire AST tree with free_ast()
- [ ] Optional: add print_ast() for debugging purposes

Use this checklist to ensure your AST is robust, traversable, and easy to debug.