

```

class ConstraintSatisfactionProblem:

    def __init__(self, variables, domains, constraints):
        self.variables = variables
        self.domains = domains
        self.constraints = constraints

    def is_consistent(self, variable, value, assignment):

        return all(
            constraint(variable, value, assignment)
            for constraint in self.constraints.get(variable, [])
        )

    def backtrack(self, assignment):

        if len(assignment) == len(self.variables):
            return assignment

        var = self.select_unassigned_variable(assignment)

        for value in self.order_domain_values(var, assignment):

            if self.is_consistent(var, value, assignment):

                assignment[var] = value

                result = self.backtrack(assignment)

                if result is not None:
                    return result

                assignment.pop(var)

        return None

    def select_unassigned_variable(self, assignment):

        return next(
            variable for variable in self.variables
            if variable not in assignment
        )

    def order_domain_values(self, variable, assignment):

        return self.domains[variable]

variables = ['A', 'B', 'C']

domains = {
    'A': [1, 2, 3],
    'B': [1, 2, 3],

```

```

    'C': [1, 2, 3]
}

constraints = {

    'A': [
        lambda var, val, ass: 'B' not in ass or ass['B'] != val
    ],

    'B': [
        lambda var, val, ass: 'A' not in ass or ass['A'] != val,
        lambda var, val, ass: 'C' not in ass or ass['C'] != val
    ],

    'C': [
        lambda var, val, ass: 'A' not in ass or ass['A'] != val,
        lambda var, val, ass: 'B' not in ass or ass['B'] != val
    ]
}
csp = ConstraintSatisfactionProblem(variables, domains, constraints)

solution = csp.backtrack({})

if solution:

    print("Solution found:")

    for variable, value in solution.items():
        print(variable, ":", value)

else:

    print("No solution found")

```

Output:

Solution found:

A : 1

B : 2

C : 3

=== Code Execution Successful ===