

```
import java.util.List;
import java.util.ArrayList;

public class QuickSort implements DivideAndConquerSort {

    @Override
    public boolean isSimple(List<Integer> problem) {
        return (problem.size() <= 1);
    }

    @Override
    public List<Integer> simplySolve(List<Integer> problem) {
        return problem;
    }

    @Override
    public List<List<Integer>> decompose(List<Integer> problem) {
        List<Integer> subProblem1 = new ArrayList<Integer>();
        List<Integer> subProblem2 = new ArrayList<Integer>();
        int pivot = problem.get(0);
        for (int i = 1; i < problem.size(); i++) {
            Integer nextValue = problem.get(i);
            if (nextValue <= pivot) {
                subProblem1.add(nextValue);
            } else {
                subProblem2.add(nextValue);
            }
        }
        List<List<Integer>> result = new ArrayList<List<Integer>>();
        List<Integer> middle = new ArrayList<Integer>();
        middle.add(pivot);
        result.add(subProblem1);
        result.add(middle);
        result.add(subProblem2);
        return result;
    }

    @Override
    public List<Integer> combine(List<Integer> problem,
        List<List<Integer>> subSolutions) {
        List<Integer> solution = new ArrayList<Integer>();
        for (List<Integer> subSolution : subSolutions) {
            for (Integer value : subSolution) {
                solution.add(value);
            }
        }
        return solution;
    }
}
```