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

public abstract class DivideAndConquerSortingMachine {

        public final List<Integer> solve(List<Integer> problem) {
                List<List<Integer>> subProblems;
                if (isSimple(problem)) {
                        return simplySolve(problem);
                } else {
                        subProblems = decompose(problem);
                }
                List<List<Integer>> subSolutions = new ArrayList<List<Integer>>();
                for (int i = 0; i < subProblems.size(); i++) {
                        subSolutions.add(i, solve(subProblems.get(i)));
                }
                return combine(problem, subSolutions);
        }

        protected abstract boolean isSimple(List<Integer> problem);

        protected abstract List<Integer> simplySolve(List<Integer> problem);

        protected abstract List<List<Integer>> decompose(List<Integer> problem);

        protected abstract List<Integer> combine(List<Integer> problem,
                        List<List<Integer>> subSolutions);

}
```