

## Algorithm Level

- **algorithm selection** (e.g., selecting the best sorting algorithm for a data set)
- **algorithm configuration** (e.g., in case-based reasoning: definitions of distance between instances, treatments to missing data, outlier removal, number of nearest neighbors to use)
- **algorithm combination** (e.g., starting with QuickSort and switching to ShellSort when a list becomes smaller than a threshold)



## Compiler Level

- **code optimizations** (e.g., number of times to duplicate a loop body in loop unrolling, tile size in loop tiling, policies in function inlining, code layout)
- **data optimizations** (e.g., data layout, tile size in array tiling, data placement in memory, structure splitting)



## Execution Level

- **task scheduling** (e.g., which tasks to run on GPU-like accelerators, job granularity, static job partition and scheduling or dynamic job stealing, number of threads)
- **resource allocation** (e.g., selection of memory allocators, selection of garbage collection algorithms)