Research issues: uniprocessor

Static branch prediction
- based on program constructs

Code scheduling for superscalar processors
- need lots of independent instructions
- speculative instructions
- VLIW compilers
- software pipelining

Integrating register allocation & code scheduling
- data structures that reflect register & functional unit interference
- heuristics, e.g., treat registers like any functional unit

Locality analysis for arrays
- prefetching
- code scheduling
- tiling

Research issues: uniprocessor

Smart interprocedural optimizations
- better application of local optimizations
- smart inlining
- whole program register allocation for global variables
- generate more efficient code in object-oriented languages

Pointer analysis
- better application of local optimizations
- for speculative execution
- to decide if two processors share data

Dynamic compilation
- apply optimizations at run-time once compute the value of invariant variables

Binary translation
- instrument code for performance metrics
- perform optimizations
- convert from x86 to some other instruction set

Research issues: parallel

Automatic parallelization of loops
- decide which loop iterations have iteration-independent data
- decide which loop iterations have data with good spatial locality
- decide which loop iterations have low inter-processor communication
- generate code to do:
  - process scheduling
  - synchronization to shared data
  - inter-process communication

Shared data restructuring
- eliminate false sharing

Research issues: parallel

What I have omitted (at least):
- debugging
- high-level language optimizations
- garbage collection
- better program analysis