Exam info:

- 90+: 1
- 80+: 8
- 70+: 7
- 60+: 4
- median: 77
- average: 76.5

Announcements:

- Assignment 3 now due 10/26 (takes a lot of time to run, start now!)
- Assignment 4 will be the last assignment, to spend more time on project.
- Critiques now due at 8AM.
- References in papers we write need to include page numbers.

NOTES:
finishing Conte paper. . .

- Def 2.5 – Directed Reference Graphs (DRG)
  - Control flow between instructions
  - Vertices are addresses
  - Edges are DRG, flow between
  - Reference Graph edges weighted, nodes also weighted.
  - Used for detecting phases in program.
- Def 2.7 – Can break graph into subgraphs for phases
  - Fine-grained phases
    - From a node, any other node in phase can be reached. In new phase, items guaranteed not to be previously referenced.
- Def 2.8 – Probability that phase is encountered.
  - Can reason about phase behavior from this.
- Weighted basic block graph - collapsing of the bigger graphs.
- Def 2.9, 2.10 – Branch prediction probability statically assessed (i.e., weights of branching edges in graph)
  - Mulitiply weighted edges by weight of node to get probability of occurrance for the branch.
  - Compilers can optimize by changing branch opcode to hint to the dynamic branch predictor about what to expect.

- Table 1:
  - Q: Why are the locality and phase behavior functions included in "flow control" table?
  - A:
    - Temporal locality: These behaviors can be correlated to control behavior.
    - Spacial locality: Indicator of control flow.
    - Phase behavior: density important for flow control.
- Data flow GRIPs:
  - Can analyze lifetime of variables → locality/reuse of variables.
  - Minimize number of registers needed.
    - But then register renaming in hardware has to reverse this action.
    - Might be good if compilers did less of this.
    - However, still need compilers to do this to some degree for ease use/analyzing.
    - Use graph coloring to determine number of registers needed.
    - Number of registers estimated by life density function.
- Def 2.11 – Variable life density function
  - Number of registers used.
  - Number of spill/fill code.
- Table 2 – Data flow GRIPs
- Def 2.12 – Data dependence behavior (Data flow graph)
  - This is related to lab 3.
  - Tight dependencies vs. loose dependencies.
  - Indication of amount of dynamical scheduling possible.

SYMPO paper

- Auto-create max-power benchmarks.
  - To test chips.
  - Discover ways to exceed specified TDP.
  - Don't want to over-design thermal for processors, so need to find a way to reach a realistic max power level.
  - Companies usually have knowledgable designer hand-write a benchmark in assembly for worst-case.
- Takes aways need for human judgement of worst-case scenario – use genetic algorithm.
- Find practically-attainable maximum.

- Need it for the whold system: core and uncore.
- Only reference tests for x86 were publicly available.
- Tests on actual hardware (AMD-designed board).
  - Objective to beat k7 previous worst-case.
- start with a random sequence of instructions → read power → genetic algorithm → another case generated to test → read power, etc.