

Feedback on 339 exam, A.Y. 2019-20. Q1-2: Giuliano Casale, Q3-4: Holger Pirk.

\* Q1

This question was answered very well by most students. The most common mistake appeared upon calculating the database utilization in 1b.ii): several students did not factor in that the database requests originated only from the History page.

\* Q2

This question was also well done, although to a lesser degree than the first question. A number of students did not really include the explicit expression of the response model in 2a.i. Many students did not get entirely right 2b.i, as they did not comment under which assumptions the AR model holds and what to do when they do not hold in a given time series. The rest of part 2b was done nearly perfectly by most, with one common issue being that some calculated the settling time using the natural logarithm, others using the logarithm base 10, and some base 2. We have accepted all these answers awarding the same points since the slides were slightly ambiguous as they only mentioned "log", but keep in mind that the correct one is the natural logarithm.

\* Q3

Generally, students struggled slightly more with this question than the others. Parts a and b were mostly answered correctly (though some students struggled with part b ii). In part c, many students forgot to model the fact that there is a traversal of the tree for *every* value in input1. Also, the discussions of alternative ways to model the program were less convincing than I had hoped for. Part d was designed to be a challenge but it seemed most students ran out of time rather than ideas.

\* Q4

This question was answered expectedly well by most students. Parts a and b were generally answered well (though some ambiguity existed in the hazards in part b which were taken into account during marking).

Part c was surprisingly challenging as many students seemed to forget about two things: first, memory bound includes any data access, including those to underlying caches (L2 and L3). Second, that L1 and L2 caches are a per-core resource that, consequently, scales with the degree of parallelism. The fact that the profile does not change (significantly) while memory bandwidth naturally becomes scarcer with more cores contending for it makes it unlikely that the application is plainly memory-bound but has to be bound by a per-core resource.

Part d was, by design challenging but answered as well as expected.