1. What is the problem with switch-based interpreter loops from a performance standpoint?

    Modern (pipelined) CPUs must be able to reliably predict what machine-code instructions should be executed next in order to achieve high performance. However, the switch statement makes it virtually impossible for a CPU to predict what instruction should be executed next (since it depends on which bytecode operation needs to be executed).

2. What is one benefit and one drawback of using tagged pointers?

    Benefit: Some types of values (e.g., integers) can be accessed directly without needing to dereference the pointer, thereby reducing the number of memory accesses.

    Drawback: Accessing values that are stored in tagged pointers require additional instructions (e.g., to negate an integer that's stored in a tagged pointer, one would have to perform a shift to remove the tag, compute the actual negation, and then perform another shift to re-tag the pointer).