Which of the following are true about the Just-In-Time compiler? 4 points

- [ ] The Just-In-Time compiler performs optimizations on your code as it is running
- [ ] The Just-In-Time compiler first optimizes methods that are invoked more often
- [ ] It is difficult to predict when optimizations occur and even what an optimization does
- [ ] The Just-In-Time compiler can have a drastic effect on a program's empirical runtime (how many seconds a program takes to run)

Peyrin and Justin are debating between asymptotic analysis or empirical * 4 points timing. Peyrin thinks that empirically timing algorithms is a better approach to understanding the runtime of algorithms. Justin thinks that asymptotic analysis is better.

In what ways is **Justin** correct?

- [ ] Asymptotic analysis helps provide a quick understanding of an algorithm's runtime, whereas timing an algorithm usually takes longer or might not be possible
- [ ] Asymptotic analysis measures the number of seconds an algorithm takes to execute better than empirical timing an algorithm
- [ ] All algorithms have an asymptotic runtime that can be easily calculated
- [ ] Asymptotic analysis helps understand how an algorithm will perform on very large inputs, which might not be realistically timed

Peyrin and Justin are debating between asymptotic analysis or empirical timing. Peyrin thinks that empirically timing algorithms is a better approach to understanding their runtimes. Justin thinks that asymptotic analysis is better.

* 4 points

In what ways is **Peyrin** correct?

☐ On smaller inputs, code with a larger asymptotic runtime may run faster in practice than code with a smaller asymptotic runtime

☐ Empirical timing measures how long an algorithm actually takes to run in seconds

☐ Optimizations from a compiler may make programs with a larger asymptotic runtime run faster in practice than code with a smaller asymptotic runtime

☐ Results from empirical timing are independent of the computer used to run the code

Submit   Clear form

Google Forms