St. Francis Xavier University
Department of Computer Science

CSCI 435: Algorithms and Complexity
Week 8 Discussion Questions
Winter 2023

[Imp95] Russell Impagliazzo. A Personal View of Average-Case Complexity. *Proc. of SCT 1995*, 134–147, 1995.

Russell Impagliazzo is a professor of computer science at the University of California San Diego. He is well-known for his work in computational complexity theory as well as for selected results in randomized algorithms, including work on pseudorandom number generation algorithms.

Read the first two sections of this article, and skim through the third section. Then, consider the following questions.

1. What are the five possible worlds of average-case complexity, and what do each of these worlds imply as outcomes for complexity theory?
2. Which of the five worlds is closest to our real world, and why?
3. Personally, which of the five worlds do you believe is most likely to arise in the (near or far) future?
4. What is the distinction between expected time and average-case time? Is this an important distinction to make?

[BS84] Andrei Broder and Jorge Stolfi. Pessimal Algorithms and Simplexity Analysis. *ACM SIGACT News*, 49–53, 1984.

Andrei Broder is a distinguished scientist at Google, and a specialist in internet technology and the web graph. Jorge Stolfi is a professor at the State University of Campinas in Brazil, researching aspects of computer vision, computational geometry, and image processing.

Read all sections of this article, following the analyses as closely as you can. Then, consider the following questions.

1. What is a "reluctant algorithm"? Why would we measure such algorithms in terms of their best-case runtime?
2. How do we define the "simplexity" of a problem? What does it mean for an algorithm to be "pessimal" for a particular problem?
3. What are the "real world" analogues of each of the previous concepts that computer scientists actually study?
4. Consider the slowsort algorithm. What paradigm does it use, and how does the algorithm itself function?
5. How does the slowsort algorithm distinguish itself from the very similar selection sort algorithm?
6. This article was purposely written to be tongue-in-cheek and to poke fun at more serious algorithm analysis papers. What was your favourite part of the article?