St. Francis Xavier University
Department of Computer Science

CSCI 435: Algorithms and Complexity
Week 2 Discussion Questions
Winter 2023

[Gar60] Martin Gardner. Mathematical Games. *Scientific American*, 202(2):150–156, 1960.

"Mathematical Games" was a famous column that ran in *Scientific American* magazine for 24 years. Its author, Martin Gardner, was a popular mathematics writer known for his interest in, and exposition of, recreational mathematics.

Read Section 3 of this article and consider the following questions.

1. Recall our formulation of the secretary problem from our lectures. How does the formulation of the problem in Section 3 compare to our formulation? That is, what are the "secretaries" in this case, how do we measure the "quality" of a choice, and so on?
2. The article notes that the best strategy "is not the same as [...] a strategy that will maximize the *value* of the selected number". Is there a true difference between these notions, or do you believe the best strategy ought to always give you the largest value?
3. Follow the advice at the end of the article: take 10 slips of paper, write varying values on each slip, and use trial and error to see whether the number of slips you reject in your experiments align with the stopping strategy we learned in lecture. Take notes on your experiments and report on how they compare to the theoretical results.

[ST85] Daniel D. Sleator and Robert E. Tarjan. Amortized Efficiency of List Update and Paging Rules. *Communications of the ACM*, 28(2):202–208, 1985.

Daniel Sleator is a professor of computer science at Carnegie Mellon University who did pioneering work in the fields of amortized complexity theory and data structures. Robert Tarjan is a professor of computer science at Princeton University and a Turing Award recipient who is well known for his work in graph algorithms.

Read the article, but feel free to skip over the proofs if you like. Then, consider the following questions.

1. Section 2 of the article defines the "dictionary problem". Which data structure do the authors use to implement their solution for the dictionary problem? What benefits might this data structure impart for this problem? Are there any drawbacks that we might be able to overcome by using a different data structure?
2. For what reasons do the authors cite as evidence that the move-to-front strategy is better than either the frequency count strategy or the transpose strategy?
3. Based on what the authors state, which strategy is the best choice if we restrict ourselves to an offline scenario?
4. What is "demand paging"? Are there any examples of paging rules you can think of that don't use demand paging? (Such rules won't necessarily be mentioned in this article.)
5. Why is the longest forward distance strategy (or MIN algorithm) not an online algorithm?
6. Later in the course, we will focus our study on amortized complexity. For the time being, consider the authors' remarks on amortized complexity in this article. Why is amortized complexity better, in a sense, than average-case complexity?