Ramel Cary B. Jamen 2019-2093

CSC 124 - Design and Analysis of Algorithms
0: D-Assessment
January 16, 2024

Context
The most straightforward reason for analyzing an algorithm is to discover its characteristics in order to evaluate its suitability for various applications or compare it with other algorithms for the same application. But:

1. What is an algorithm? If you rate yourself from 1 to 10 of how well do you know the algorithm in general, what it would be and why?
2. Have you ever analyzed an algorithm? In what way? Discuss.
3. What algorithms do you think should be preferred, and what should be avoided?

Instruction
Answer the 3 questions above. Submit in this classroom before the deadline.

Answers

1. When talking about algorithms, the first person that comes to my mind is Donald E. Knuth, a famous computer scientist and the author of *The Art of Computer Programming* (TAOCP), a highly recommended book for anyone eager to delve into programming. I liked the algorithm definition that I've read in his first book (Fundamental Algorithms) where he breaks it down into 5 basic properties: Finiteness - an algorithm must start and always terminate after a finite number of steps. Definiteness - each step of an algorithm must be precise and unambiguous, it should not be open to multiple interpretations. Input - an algorithm has zero or more inputs (that can also be served as states of the program). Output - an algorithm has one or more outputs that have specified relation to the input(s). Effectiveness - an algorithm is said to be effective if it is sufficiently basic that a person can express using a pencil and paper. Those 5 properties I think are critical when creating a systematic solution to a problem. Rating myself would be an honest 6. With continuous learning, I somehow developed an algorithmic thinking ability that allows me to think sequentially and in a stateful manner (e.g., MVC Model in Web Development, States in Javascript, Recursion in Functional Programming).

2. Yes, in my CSC145 Programming Languages subject, I have experienced describing syntax and translating semantics (denotational and operational) of a given program into simpler language and abstract machine instructions. It basically revolves around how a program works and behaves in a stateful manner. Also when working with a collaborative project, I also need to read other people’s code as a progress to the features and goals of the project.

3. Personally, in making an algorithm, it must satisfy the 5 properties of algorithm by Donald Knuth, or given that the algorithm is efficient by runtime, inexpensive in resources, with clarity, and coherent. And that it is achievable by building a good foundation in Data Structures, Modelling Design, and experience working in the industry by reading codes, and learning coding conventions. Avoid writing inappropriately complex algorithms because it tends to be slow, costly, and prone to bugs (e.g., infinite loops, race conditions, overflows).

References

Knuth, D. (1968). *The Art of Computer Programming*. Www-Cs-Faculty.stanford.edu. https://www-cs-faculty.stanford.edu/~knuth/taocp.html