Computer Science Education: Synthesis of the Curricula of Data Structures and Algorithms

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Mark Chitre
Spring, 2022

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Rosanne Vrugtman, Department of Computer Science

Computer Science Education: Synthesis of the Curricula of Data Structures and Algorithms

CS4991 Capstone Report, 2022

Mark Chitre
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
mnc8rj@virginia.edu

Abstract
Computer Science (CS) education is constantly evolving and curricular design can actually limit learning and work-related knowledge. One solution is optimizing CS coursework to help students learn practical skills in a way that trains them for software development and auxiliary fields in CS. The main way this can be accomplished is through the synthesis of the first data structures course and a high-level algorithms course. This synthesis is achieved by intertwining assignments and learning through these curricula to form a course in which students can use one topic to facilitate learning of another. An example of this is implementing a search algorithm with a specific data structure that is required in an application of learning in another course or assignment. The major advantage of this synthesis is to prepare students for coding interviews and assessments and jobs after graduation. Additional testing and data-gathering would be necessary in order to evaluate overall effectiveness of this approach.

1. Introduction

From the advent of computer science there has been constant advancement in how it is taught. The methodologies highlight different aspects of computer science and they also differ in the sequence of ideas that are taught. Major concepts such as data structures, algorithms, architecture, and testing techniques are all taught throughout general CS curriculum with the order being altered for different institutions. Designing curriculum involves looking at how these concepts interact with each other and then creating a series of courses with each successive course used as a stepping stone for the next. As a result, the way that concepts are prioritized can be subjective based on who is designing the curriculum and for what purpose.

2. Background

Two major concepts of computer science curricula are data structures and algorithms. In CS, there is a necessity to store data optimally so that it can be processed later. At the most elementary level, this can be done using a variable, which associates a value with a variable name and puts that association somewhere in memory. Data structures allow for multiple values to be stored with one associated variable name, and there are a plethora of structures that fit different uses well. For example, using a data structure called a Binary Tree will break the data down into halves using a parent node, which has two child nodes under it. One of the strengths of a Binary Tree is to apply recursion (calling a method/process on itself) to process data.

Algorithms, a set of instructions a computer follows, are written by developers to complete a process. They are sorted into many general groups based on what they achieve. An example of an algorithm is a Binary Search, which will find the target value specified based on its parameters. Algorithms make data manipulation and processing a great deal faster and easier because they can be used rather than specifying code for every situation. In order to write the majority of algorithms, data structures are required to store and manipulate data; and for all the different algorithms there are some data structures work better than others.

3. Related Works

The research on CS pedagogy about the topics of data structures and algorithms goes into depth about their combination in teaching. North Carolina State University recently redesigned its second CS course to include projects for the purpose of mastering interactions between theory, data structures and applications to algorithms (King, 2021). This also includes the students conducting experimental tests between implementations of different data structures to understand which algorithms and data structures fit together best (King, 2021).

The response to this change in curriculum was overall very positive, with 90.6% of students saying the project helped with the ideas of analysis and design in software development (King, 2021). This relates to the idea that there is a benefit in combining major topics, in this case data structures and algorithms, because it provides a more-than-sufficient level of student understanding about software development. Additionally, by learning to experiment with different data structures combined with algorithms, a super-majority of students found the project helpful.

Tenenberg (2003) asserts that data structures and algorithms should also be taught together with a different type of progression. This framework approach has students use the STL (Standard Template Library) for algorithms at first to master the use of data structures, then implementing algorithms using the learned data structures. This framework supports the combination of the two topics in a different method than in the previous paper but it still favors the teaching of the ideas of data structures and algorithms together. The benefits of this framework are that for post-graduation work or jobs it is important to understand when to use included CS libraries and what to change if alteration is needed (Tenenberg, 2003).

These two papers both support the combination of data structures and algorithms but in different ways and show the advantages of the synthesis.

4. Project Design

The new curriculum will follow a similar structure to project-based classes with focus on the advantages/disadvantages of using certain data structures with certain algorithms.

4.1. Overview of Design

A prospective new curriculum includes several takeaways from previous implementations of other institutions such as projects to explore the advantages for the use of multiple data structures for an algorithm. The mastery of these assignments promotes job-specific skills that will benefit students through a new learning method. The curriculum will also not only teach simple data structures with simple algorithm but will mix the difficulty of these concepts to promote mastery of the content.

4.2. Content Delivery

The content of the curriculum would follow a general theme of teaching an algorithm, implementing it with different data structures, and then comparing the performance/advantages of each data structure taught for the algorithm. An example of this is first using arrays to master a binary search, then moving onto using lists for the same operation. As shown in Figure 1, the course will start with primitive data structures and move on to user-defined and built-in compound data structures.

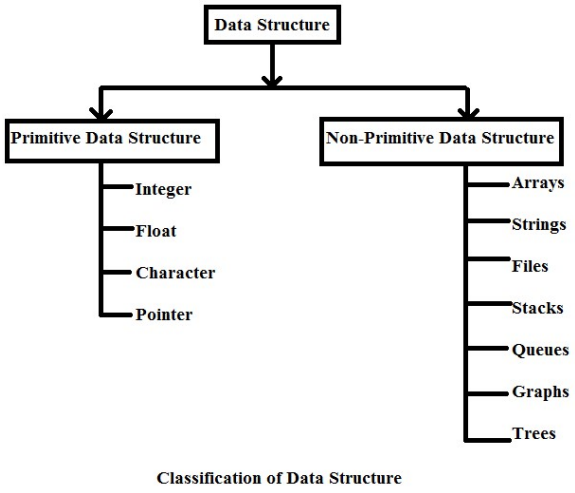

Figure 1: Breakdown of data structures (Nakanyike, 2018)

Content would be delivered through projects (for the more complex algorithms), assignments and homework, as well as exams. Some content also directly builds off other content. In order to learn about algorithms in graphs, the student will also have to explore what exactly graphs are and what they are used for, and that content will be taught sequentially with follow-up assignments that will pursue both sections of the topic.

Shown in Figure 2, there are graphs with distinct properties that to be mastered before one can apply algorithms to parse through graphs. The properties of these graphs will be taught before they can be analyzed with algorithms. This content also be delivered dynamically with content from the first couple weeks also being used heavily toward the end of the course.

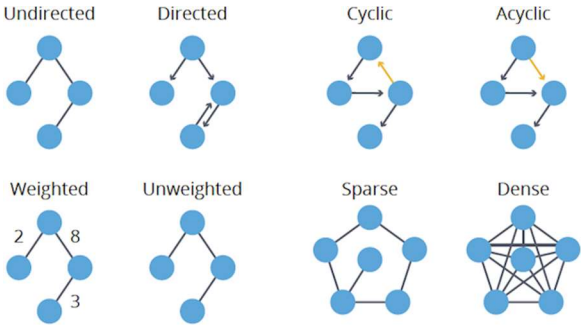

Figure 2. Types of graphs (Needham, 2018)

4.3. Challenges

Delivering CS curriculum in this way could make grading difficult because of nature of teaching data structures. Students that do not master specific elementary data structures, no matter the reason, may face double jeopardy in grading as more algorithms are introduced. For example, if a student lacks proficiency in using arrays, resulting points taken off assignments will compound and this may amount to the student getting a far lower grade than others. There are several ways to avoid this in grading by scaling grades or creating exceptions in the point system. This is the main challenge that arises with grading, but adjustments or required make-ups can prevent this.

5. Anticipated Results

This prospective new curriculum will have many benefits because of the new way that the content will be delivered. Combining data structures and algorithms will not only benefit students in future CS classes but also shift their method of learning so that they will be more prepared for job interviews and tasks that jobs assign.

The redesigned North Carolina State University curriculum was taken well by students. 202 of 536 (37.7%) of students

responded to the survey asking if the new curriculum helped with understanding of analysis and design during software development, 90.6% percent responded yes to this (King, 2021). Using a similar method of content delivery, the anticipated results to this change will be similar to those of the previously mentioned study.

Additional results, such as preparedness of students for specific job tasks will be very hard to gauge because a lot of combined factors contribute to that job-preparedness. Teaching the curriculum using a synthesis of these major topics is anticipated to add several takeaways from the courses that students have previously not gotten.

6. Conclusion

Implementing this curriculum will lead to benefits throughout the rest of a student's higher education and extend to the workforce, as well. The importance of this project is that it optimizes the way computer science is taught in order to promote better overall understanding for students. Including data structures and algorithms in a combined manner allows for the improved delivery of content and will prepare students for interviews, as well.

This content will be taught through homework assignments, projects, and exam so that complex algorithms and data structures can work off of the ideas of the other. Based on the results of similar implementations, it is expected that this proposed curriculum will help with understanding analysis and design. There is great potential because of this change which is likely to benefit students immensely in similar applications.

7. Future Work

There is significant future work to be done after this curriculum implementation is finished, mainly including complex evaluation and analysis on the effects of the new curriculum. Further adjustments can be made as a direct result of this, gauging what has worked and what has not.

Future work can also address greater specifics of the curriculum that have just been outlined, such as a more detailed construction of the interaction between specific data structures and algorithms. There are more major interactions between topics in CS that can also be explored and added into another class that would be taken before or after the topics in the proposed curriculum.

References

King, J. (2021, March 1). *Combining theory and practice in Data Structures & Algorithms Course Projects: Proceedings of the 52nd ACM technical symposium on computer science education*. ACM Conferences. Retrieved February 22, 2022, from https://dl.acm.org/doi/10.1145/3408877.3432476

Mariam, N. (2018, September 26). *The importance of algorithms and data structures in our day to activities*. Medium. Retrieved March 12, 2022, from https://medium.com/@mariam.nakanyike/the-importance-of-algorithms-and-data-structures-in-our-day-to-activities-7a9dac119725

Needham, M., & Hodler, A. E. (2019, April 8). *Graph algorithms in neo4j: Graph algorithm concepts*. Neo4j Graph Data Platform. Retrieved March 12, 2022, from https://neo4j.com/blog/graph-algorithms-neo4j-graph-algorithm-concepts/

Tenenberg, J. (n.d.). *A framework approach to teaching data structures*. Washington University. Retrieved

February 22, 2022, from https://faculty.washington.edu/jtenenbg/publications/tenenbergDataStructuresFrameworks-sigcse2003.pdf