# BOARD # 85: WIP: BST Cards: A Tangible Binary Search Tree (BST) Activity for Developing Algorithmic Thinking in Middle School Students

**Duncan Johnson, Tufts Center for Engineering Education and Outreach**

Duncan Johnson is an undergraduate student at Tufts University majoring in Computer Science. He is the co-founder and Executive Director of BX Coding, a STEM education nonprofit dedicated to building open-source software and curriculum for STEM educators. Through his work with BX Coding and the Tufts Center for Engineering Education and Outreach (CEEO), his research has focused on educational methodologies and technologies that introduce elementary and middle school students to Computer Science. Currently, he's exploring how technologies can leverage generative AI to better support educators.

**Dr. Ethan E Danahy, Tufts University**

Dr. Ethan Danahy is a Research Associate Professor at the Center for Engineering Education and Outreach (CEEO) with secondary appointment in the Department of Computer Science within the School of Engineering at Tufts University. Having received his graduate degrees in Computer Science and Electrical Engineering from Tufts University, he continues research in the design, implementation, and evaluation of different educational technologies. With particular attention to engaging students in the STEAM content areas, he focuses his investigations on enhancing creativity and innovation, supporting better documentation, and encouraging collaborative learning.

**Elliot Benjamin Roe, Georgia Institute of Technology**

# WIP: BST Cards: A Tangible Binary Search Tree (BST) Activity for Developing Algorithmic Thinking in Middle School Student


## Abstract

Introducing algorithmic data structure concepts to middle school students poses a unique challenge due to their complexities and reliance on code-specific syntax. Aiming to simplify this process, we adapted a tangible Binary Search Tree (BST) activity from the CS Unplugged curriculum to promote an experiential understanding of the BST data structure. This paper presents our iterative approach to developing this activity, first through describing our preliminary qualitative findings and then by proposing a mixed-method study to formalize our inquiry with quantitative data.

We implemented two versions of the BST activities in week-long summer coding camps for middle school students during June and July 2023. Our qualitative observations and field notes from these implementations informed future design considerations. In our first iteration, students drew BSTs with chalk on pavement, and they struggled with the inflexibility of chalk diagrams. In our second iteration, students created BSTs with manipulable notecards and physical connectors made from string, and a spy-themed narrative was added to increase student engagement. In this second iteration, we observed that students often bypassed systematic tree traversal by visually scanning for targets, potentially undermining conceptual understanding of BST efficiency.

Building on our qualitative findings, we propose a mixed-method study of middle school students in informal, summer STEM learning environments. Pre- and post-surveys will examine student self-reported engagement and enjoyment. An activity quiz will assess if students can identify correctly-structured BSTs and perform simple operations on a BST. Interviews with implementing educators will provide information on student behavior, questions, and misconceptions during the activity.


## Introduction

A Binary Search Tree (BST) is a powerful data structure for storing and interacting with ordered data. Search, insertion, and deletion in a BST can be significantly more efficient than linearly-ordered data [1]. To fully teach students about BSTs, instructors must explain complex concepts such as recursion, which have been shown to overwhelm middle school students, especially if taught in a manner that includes implementation details and syntax [2] [3].

While CS education for younger students often focuses on fundamental programming syntax, we believe it's important for young students to engage with a vast array of core computing concepts and skills in addition to learning programming language syntax. Our goal is to leverage the visual representation of BSTs so students learn the core concept that the intentional structuring of data can yield algorithmic efficiency improvements. Rather than diving into a BST's implementation in code, students tangibly interact with BSTs away from their computers.

This paper is structured in two parts. First, we present qualitative findings from our initial BST activity implementations with middle school students. Then, we outline a proposed mixed-method study design that will formally assess the efficacy of our refined BST activity. Our preliminary work serves as the foundation for this more rigorous investigation.

## CS Unplugged

Developed by researchers at the University of Canterbury, CS Unplugged is a collection of activities designed to teach children ages 5-14 about computer science topics without using any computers [4]. For classrooms that normally use computers to teach and practice computer science, CS Unplugged activities give students a break from typing at their computer and let them interact with their classmates face-to-face. Unplugged activities also give students the opportunity to tangibly engage with computer science concepts without the distraction of an Integrated Development Environment (IDE) interface or the challenges of typing, spelling, and debugging.

We take inspiration from CS Unplugged's "Binary search trees" activity[1], which is designed for ages 11-14 and is estimated to take 45 minutes [5]. CS Unplugged's BST activity plan has the instructor draw a BST of numbers on the ground, then cover each of the numbers with a disc so that only one number can be seen at a time. Students are then asked to search for numbers by starting at the root of the tree before traversing down the tree, revealing one number at a time by picking up the disc covering that number. The activity also provides several variations for an instructor to add to their lesson, including adding a value to the tree, building your own tree, and building a tree with words.

## Preliminary Qualitative Study: Chalk BST

We began our preliminary qualitative study by implementing a simple adaptation of the CS Unplugged activity on BSTs in a week-long coding-focused summer camp during June 2023, with 11 middle school students in attendance. The instructors collected qualitative data through field notes and observations of student interactions with the activity and materials. Instructors then conducted a discussion at the end of the day to analyze these observations.

This preliminary study had several limitations including instructor bias (they adapted and designed the activity themselves), the small sample size (11 students), and potential selection bias from the paid summer program format. We acknowledge that these factors limit the generalizability of our findings, which should be interpreted as exploratory insights rather than

[1] https://www.csunplugged.org/en/topics/data-structures-for-searching/binary-search-trees/

definitive conclusions.

After a short presentation about BSTs from an instructor, the class moved outside and students were divided into groups of 2-3. Each team was asked to build their own BST of words. An instructor had a list of 20 different words and would give groups one word at a time to insert into their BST. Ordering the words alphabetically, students would add a word to their BST by drawing with chalk on the pavement. Figure 1 is an example of a chalk BST.

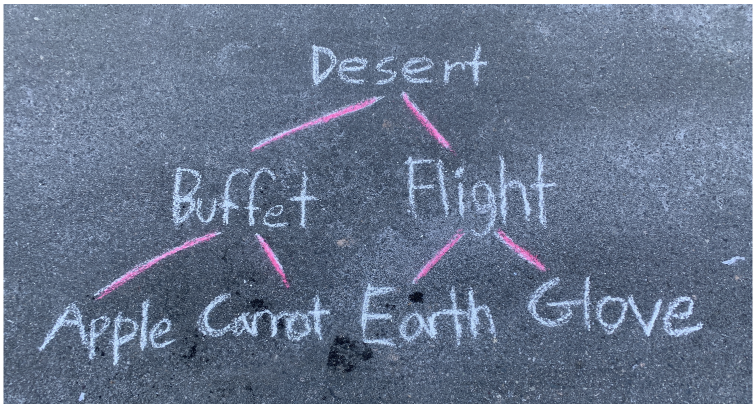



*Figure 1.* A Binary Search Tree drawn on the pavement with chalk.

### Chalk BST Qualitative Findings

This form of the activity was challenging because of the inflexibility of the chalk. Students had to draw the BST node-by-node, which meant that finding space to draw new nodes inside an increasingly cluttered BST was difficult for students. This led to disorganized trees that caused frequent mistakes. When these mistakes occurred, it was difficult to fix them because erasing the chalk from the pavement was challenging.

### Activity Iteration: BST Cards

Based on our qualitative findings from the first implementation, we sought to improve the activity for a second week-long coding camp in July 2023 with 15 middle school students. The CS Unplugged activity recommends writing data on discs or cards and placing them in a BST arrangement [5]. We adapted this by adding two physical connectors made out of string and alligator clips to the bottom of each notecard. The relationships between data in the BST were communicated by cards being clipped into the left and right strings extending from the bottom of each card. Figure 2 is an example of a BST using notecards and clips.

To increase student engagement and highlight the power of storing data in a BST, we added a spy-themed narrative and a data-lookup efficiency competition to the activity. The activity's motivation was that an international spy network had a list of "codenames" for friendly spies and

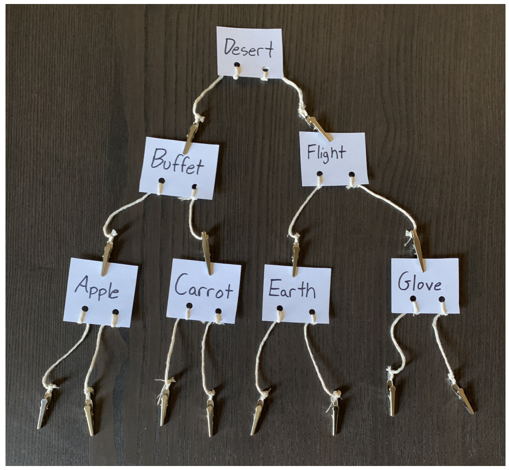



*Figure 2.* A Binary Search Tree created with cards connected by string and alligator clips.

they needed an efficient way to check if a codename was friendly or not. Each group of students constructed a BST with roughly 20 cards, each with a codename on it, by inserting the codenames one-by-one into the BST. Next, the instructor would call out codenames, with some being words present in the BST (friendly codename) and some being absent from the BST (unfriendly codename). Students were asked to search down from the root and record how many different cards they examined while determining whether the code name was friendly or unfriendly.

To contrast the efficiency of BSTs with linearly-sorted data, an instructor constructed a simple alphabetically-sorted linked list. Another instructor presented a challenge to the students: he was going to call out 10 codenames and each team had to compete against the instructor with the linked list to see who examined the fewest cards while determining whether those 10 codenames were friendly or unfriendly. In some instances, the instructor's linked list outperformed the students' BSTs, but in the vast majority of cases, the BSTs were much more efficient. After the 10 codenames were evaluated, each group added up the number of cards visited and found that their BSTs vastly outperformed the instructor with the linked list.

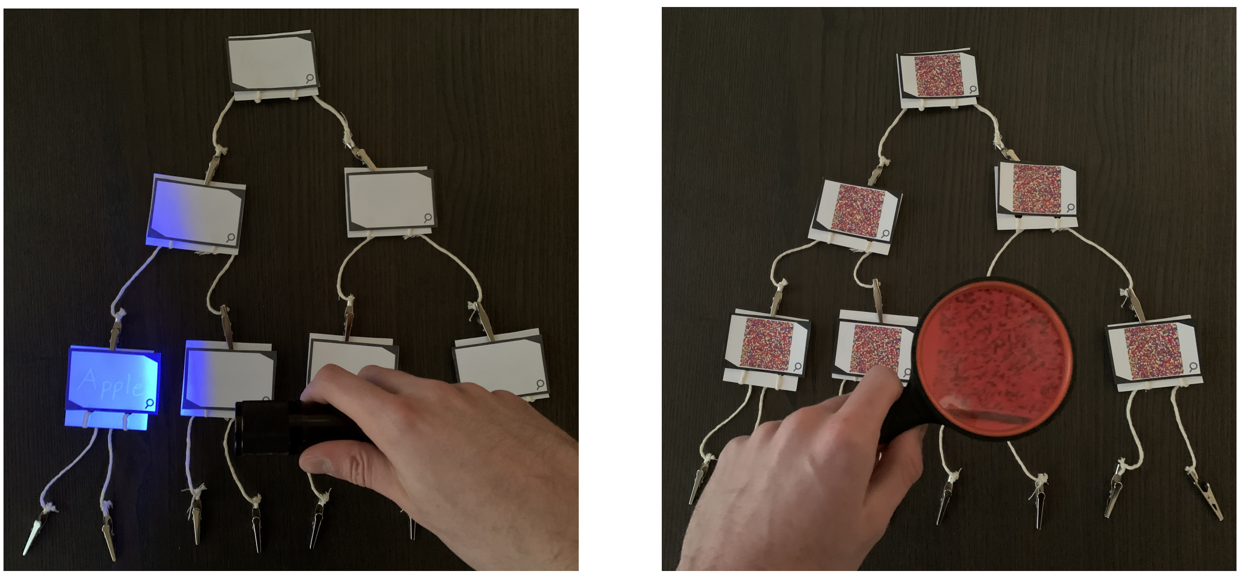
**BST Cards Qualitative Findings**

Our observational data from the second implementation suggested that the improvements made after the first implementation were successful. From our qualitative observations, students found the card-and-clip construction of BSTs easier than drawing with chalk because it was much easier to reposition cards when needed.

We noted that students were more engaged with the second implementation, suggesting that our spy-themed narrative and searching efficiency competition added to the educational impact of the activity by motivating students and providing a plausible scenario where algorithmic efficiency was important.

### Activity Design Improvements

Our qualitative observations revealed that while constructing and searching in the tree, some students simply scanned the cards to quickly locate words within the structure rather than starting at the root and traversing down the BST like a computer would. The CS Unplugged activity nudges students towards assuming the role of a computer by hiding the data under discs and asking students to flip over one disc at a time. However, with the physical connectors between cards, flipping over cards would disrupt the tree. To solve this problem in future iterations, we have two ideas that align with the spy-themed narrative of the activity. First, we write on the cards with invisible ink and give students a UV flashlight to reveal the writing on cards one at a time, as shown in Figure 3 (a). Second, we generate secret anaglyph messages that obscure words in a sea of colors [6]. Students will then reveal data by looking at a card through a red-tinted magnifying glass, as shown in Figure 3 (b). Another planned improvement is to replace the string connectors with rigid, hinged connectors.




*(a)* Shining a UV light reveals the word ”Apple” written in invisible ink on the card.

*(b)* Looking through a red-tinted magnifying glass reveals the word ”Earth” from the sea of colors.

*Figure 3.* Future ideas for obscuring all but one card at a time.

To extend the “unplugged” nature of the activity, we plan to use physical tasks to reinforce the cost of inefficient searching. If the activity took place outside, rather than simply counting how many cards were examined during each search operation, participants could be asked to run and touch a wall every time they examine a card. Students using a BST would need to run back and forth a few times per search, whereas an instructor using a linearly-ordered data structure would have to run back and forth many times. We hypothesize that by making each search more difficult, students will internalize the efficiency improvements of a BST over a linearly-ordered data structure as they gleefully watch their teacher run back and forth many times.

These design improvements will be incorporated into the activity version used in our proposed mixed-method study. Our preliminary qualitative findings, while valuable for activity refinement, highlight the need for more rigorous assessment of learning outcomes and activity effectiveness. The following section will outline our proposed mixed-method study methodology to formalize our inquiry into this BST activity's efficacy.

**Proposed Mixed-Method Study**

Our goal for a mixed-method study examining qualitative and quantitative data is to understand how this activity impacts students in three ways: engagement, comprehension, and extension to broader computational thinking ideas. To measure each of these, we have three overarching research questions:

- To what extent does the BST Cards activity engage middle school students in an informal learning environment?
- How does participation in the BST Cards activity affect students' comprehension of BST concepts and skills?
- In what ways does learning BST concepts through the BST Cards activity contribute to a broader understanding of data structures and algorithmic efficiency?

The study will include several classes of middle school students in informal computing education environments, most likely in summer or after-school programs. Since most students will opt into these programs, this will likely create a population that is more experienced and excited about computing than a general middle school classroom. Each educator will give a short presentation to their class about BSTs and then explain the spy-themed narrative. First, groups of students will create a BST by inserting codename cards one by one into their tree. Next, an instructor will explain that they created a simple linearly-ordered list of cards. Finally, the instructor will call out random codename that could be in the BSTs or not, and each team needs to traverse the BST to say whether the codename is valid or not. Depending on the class location, a physical searching penalty such as running to touch a wall could be added. Students, with their BSTs, would compete against the instructor's linearly-ordered list to find the codenames in the fewest searches.

To measure student engagement, a post-survey will ask students questions about enjoyment and interviewed instructors will be asked whether their students were engaged. To assess student comprehension, each student will complete a pre- and post-quiz that will ask them to determine if a BST is valid and where a new element would be inserted into a tree. To understand students' broader understanding, the instructor will have a class discussion about applications of data structures and the important of algorithmic efficiency, and the instructor will be interviewed on their students' thinking. The question of broader understanding is the most difficult to measure, but it attempts to study the core aim of this activity: teaching computational thinking. Understanding how to insert an element into a BST isn't inherently valuable for middle school students, but understanding why we develop intentional data structures and create efficient algorithms is beneficial for future computational learning.

## Acknowledgments

Thank you to Ava Wandersleben for co-teaching this activity, providing feedback, and supporting curriculum development.

Thank you to the Laidlaw Scholars program at Tufts University for supporting this research.

## References

[1] P. Morin, *Open Data Structures*, en, 0.1Gb. [Online]. Available: https://opendatastructures.org/ods-python.pdf (visited on Jan. 15, 2025).

[2] D. M. Kurland and R. D. Pea, "Children's Mental Models of Recursive Logo Programs," en, *Journal of Educational Computing Research*, vol. 1, no. 2, pp. 235–243, May 1985, Publisher: SAGE Publications Inc. [Online]. Available: https://doi.org/10.2190/JV9Y-5PD0-MX22-9J4Y (visited on Jan. 15, 2025).

[3] R. McCauley, S. Grissom, S. Fitzgerald, and L. Murphy, "Teaching and learning recursive programming: A review of the research literature," *Computer Science Education*, vol. 25, no. 1, pp. 37–66, Jan. 2015, Publisher: Routledge _eprint: https://doi.org/10.1080/08993408.2015.1033205. [Online]. Available: https://doi.org/10.1080/08993408.2015.1033205 (visited on Jan. 15, 2025).

[4] T. Bell, J. Alexander, I. Freeman, and M. Grimley, "Computer science unplugged: School students doing real computing without computers," *New Zealand Journal of Applied Computing and Information Technology*, vol. 13, no. 1, pp. 20–29, 2009.

[5] *Binary search trees - CS Unplugged*. [Online]. Available: https://www.csunplugged.org/en/topics/data-structures-for-searching/binary-search-trees/ (visited on Jan. 15, 2025).

[6] J. Vančura, *Secret anaglyph message*, en. [Online]. Available: https://www.khanacademy.org/computer-programming/secret-anaglyph-message/5395038998921216 (visited on Jan. 15, 2025).