Project Proposal: Educational van Emde Boas Tree Implementation

By: Eric Adams and Skylar Howe

April 3, 2020

Summary

Trees are one of the cornerstone data structures in computer science. Even the base full binary tree structure has a desirable height bound and runtime of O($\log_2 n$). There are various other tree structures that are modified from binary trees to better solve the particular problem, such as altering insertion, search, or deletion techniques. One of the more versatile modified tree structures is the Van Emde Boas tree, or vEB trees. vEB trees are unique since they are "always within a factor of the best tree". Meaning that, even though it may not be the most optimal tree for the job, it is fairly close to the optimal. Gaining a comprehensive understanding of this structure, its strengths, and its weaknesses is beneficial for programmers of every level. That being said, trying to conceptualize this structure is difficult. While there are plenty of great lectures to watch or papers to read, there are not many interactive or visual education tools, which often helps with solidifying concepts. Our proposal is to contribute to the greater education of the general Van Emde Boas tree data structure through implementation and creative means.

Known Results

There are numerous papers on the theoretical possibilities of vEB trees. For example, one paper written by Hao Wang and Bill Lin proposed an altered vEB tree for high performance priority queues associated with network applications. Those who are in intermediate-level programming courses could benefit from using the suggested methods. There are also some implementations that exist. There are also some implementations floating around, such as a C++ implementation developed by Yash Aggarwal. While these are useful, there are also no educational tools out there like the ones we are considering.

Potential Plan of Attack

Our idea is to provide a more creative representation of the tree, which can make it easier for students to understand the data structure by having some interactive option where they can test the data for themselves. So far, we have two concepts to consider, both of which start out the same way.

Idea #1: Implementation and GUI

We would first implement the vEB data structure in a language of our choice, and provide a program in which to test the implementation. We would then create a GUI which uses the implementation to visualize the vEB tree, and goes through a step-by-step process of the different functions. The GUI would have different amounts of data so users can go through different types of examples.

Idea #2: Implementation and Video

Again, we would start with creating an implementation. We would then use this implementation in an educational video which goes over the data structure. After explaining the general background of the data structure and theoretical ideas, a few examples with different amounts of data would be walked through during the video.

Both of these options are feasible, however our desire is to aim for Idea #1. If, however, we run into complications with creating the GUI, we would then move to Idea #2.

Works Cited

Aggarwal, Yash. “Van Emde Boas Tree.” *OpenGenus IQ*, OpenGenus IQ, 10 Mar. 2019, https://iq.opengenus.org/van-emde-boas-tree/.

Wang, Hao, and Bill Lin. “Pipelined Van Emde Boas Tree: Algorithms, Analysis, and Applications.” *IEEE INFOCOM 2007*, 2007, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.167.3128&rep=rep1&type=pdf.