

# Lab 6

CMPT 432 - Spring 2023 | Dr. Labouseur

Josh Seligman | [joshua.seligman1@marist.edu](mailto:joshua.seligman1@marist.edu)

March 27, 2023

## 1 CRAFTING A COMPILER

The two data structures most commonly used to implement symbol tables in production compilers are binary search trees and hash tables. What are the advantages and disadvantages of using each of these data structures for symbol tables?

The primary advantage of using a binary search tree for a symbol table is that it is only bounded by the size of memory, assuming the nodes are connected through objects and pointers rather than a matrix or adjacency list. The downside for binary search trees is that they have a lookup and insertion times of  $O(\log_2 n)$ , where  $n$  is the number of symbols in the tree. On the other hand, hash tables are extremely efficient with lookup times and insertion times with a time complexity of  $O(1) + \alpha$ . This  $\alpha$  is the extra time it may take to probe or iterate through a chain in the case of collisions. Hash tables with chaining, similar to trees, are not bound by a fixed allocation because of the use of a linked list. However, hash tables that use probing to handle collisions are restricted to the length of the allocated array, which may be too small to store all variables in a given scope.

## 2 DRAGON