

Lab 6

Andrew Hatch

Andrew.Hatch1@Marist.edu

April 9, 2024

1 CRAFTING A COMPILER EXERCISES

1.1 CHAPTER 8.1

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?

Advantages of Binary Search Trees:

- Simple and easy to implement, which makes them well known and widely used.
- Fast and neatly ordered trees - can find anything in $O(\log n)$ time
- Dynamic insertion; elements can be freely added or removed at will

Disadvantages of Binary Search Trees:

- Worst-case scenarios for lookup can take $O(n)$ time
- Need to be balanced unless you want to risk having a much slower runtime (like above)
- Need more memory than hash tables to store their values
- Poor performance in large data sets

Advantages of Hash Tables:

- Can be performed in constant time (given a sufficiently optimized hash table)
- More space and time efficient than BSTs, especially when dealing with large data sets
- Can be used to store any kind of data

Disadvantages of Hash Tables:

- Inefficient when there are many collisions
- Limited, static capacity
- Does not maintain the order of its elements

How good are they compared to each other?

Overall, BSTs and hash tables serve different purposes - BSTs are good for small, efficient searching and organizing, while hash tables are more useful for time efficiency or having a constant speed per search.