CS-300-10147-M01 DSA: Analysis and Design 2025 C-3

8-1 Journal

[CHANWOO BOK]

[[chanwoo.bok@snhu.edu](mailto:chanwoo.bok@snhu.edu)]

Southern New Hampshire University

### **Big O Analysis**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Structure** | **File Reading & Parsing** | **Course Object Creation** | **Overall Runtime** | **Memory Usage** |
| **Vector** | O(n) | O(n) | O(n) | O(n) |
| **Hash Table** | O(n) | O(1) per insert → O(n) | O(n) | O(n) |
| **Binary Search Tree** | O(n) | O(log n) per insert → O(n log n) | O(n log n) | O(n) |

* **File reading and parsing** occurs once per course, which is O(n).
* **Course object creation** involves storing course data in a structure:
* Vector appends each object: O(1) per insert → O(n)
* Hash Table inserts by key: O(1) per insert → O(n)
* BST inserts in order: O(log n) per insert → O(n log n) in average/worst case (if balanced)

### **Advantages and Disadvantages**

#### **Vector**

* **Advantages:**
  + Simple to implement and understand
  + Maintains insertion order
* **Disadvantages:**
  + Requires additional sorting for ordered output
  + Searching for a specific course is O(n)

#### **Hash Table**

* **Advantages:**
  + Fast lookup by course number (O(1))
  + Efficient memory use
* **Disadvantages:**
  + No inherent order — extra step required for alphanumeric sorting
  + Collisions may degrade performance in worst case

#### **Binary Search Tree (BST)**

* **Advantages:**
  + Naturally maintains sorted (alphanumeric) order
  + Allows in-order traversal to print sorted course list without extra sorting
* **Disadvantages:**
  + More complex to implement
  + Unbalanced trees can degrade to O(n) performance if not managed

### **Recommendation**

Based on the advisor's requirements and the runtime analysis, I recommend using the **Binary Search Tree (BST)** as the data structure for implementation. The advisor explicitly requests the ability to print a list of all computer science courses in **alphanumeric order**, which is directly supported by the in-order traversal of a BST. While the hash table offers better average-case search performance, it does not maintain any order, making it less efficient for sorted output. The vector is simple but inefficient for large datasets when frequent lookups or sorting are needed. Therefore, the BST offers the best balance between performance and functionality, especially for a system that needs to support both efficient data access and naturally ordered outputs.