Runtime Analysis:

Let's analyze the runtime of each data structure based on the provided pseudocode for reading the file and creating course objects.

Vector Data Structure:

Reading the file and creating course objects: O(n), where n is the number of courses in the data structure. Each line is processed once, and the insertion into the vector takes O(1) time.

Hash Table Data Structure:

Reading the file and creating course objects: O(n), where n is the number of courses in the data structure. Each line is processed once, and the insertion into the hash table takes O(1) time on average.

Tree Data Structure:

Reading the file and creating course objects: O(n log n), where n is the number of courses in the data structure. Each line is processed once, and the insertion into the binary search tree takes O(log n) time on average.

Advantages and Disadvantages of Each Structure:

Vector Data Structure:

Advantages: Simple implementation, constant-time access to elements by index, maintains insertion order.

Disadvantages: Searching for a course takes linear time, not suitable for efficient search operations when the number of courses is large.

Hash Table Data Structure:

Advantages: Efficient lookup time on average (O(1)), suitable for fast access to course information using course titles.

Disadvantages: Additional overhead for collision handling and hash table resizing, does not maintain the insertion order.

Tree Data Structure:

Advantages: Efficient search time in a balanced tree (O(log n)), maintains elements in sorted order, suitable for ordered data.

Disadvantages: Slightly more complex implementation, insertion and deletion operations may require tree balancing, which could take logarithmic time.

Recommendation:

Based on the Big O analysis and the advantages and disadvantages of each data structure, the Hash Table data structure is the most suitable choice for this application. It provides constant-time access for course lookup on average, which is important for quickly retrieving course information based on the course title. Additionally, the insertion and search operations are generally efficient, even with a large number of courses. The vector data structure would be inefficient for course search, and the tree data structure may have slightly more overhead and complexity, making the Hash Table the better choice for this scenario.