# Pseudocode and Runtime Analysis

## 1. Pseudocode

### File Handling and Data Parsing

Pseudocode for all data structures:

1. Open the file containing course data.  
2. For each line in the file:  
 a. Parse the line to extract course details.  
 b. Check for formatting errors. If errors exist, log them and skip the line.  
 c. Create a Course object with the parsed data.  
3. Add the Course object to the appropriate data structure.

### Menu Design

Pseudocode for menu options:  
1. Display the following options to the user:  
 a. Option 1: Load file data into the data structure.  
 b. Option 2: Print an alphanumerically ordered list of courses.  
 c. Option 3: Print the title and prerequisites of a specific course.  
 d. Option 9: Exit the program.  
2. Execute the action corresponding to the user's choice.

### Sorting and Displaying Courses

Pseudocode for sorting and displaying courses:  
1. Retrieve all courses from the data structure.  
2. Sort courses by course number in alphanumeric order.  
3. Print the sorted list of courses.

## 2. Runtime Analysis

The table below shows the runtime analysis (Big O notation) for each data structure:

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Vector | Hash Table | Binary Tree |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| File Reading and Data Parsing | O(n) | O(n) | O(n) |
| Adding Course Objects | O(1) (amortized) | O(1) | O(log n) |
| Retrieving Course Information | O(n) | O(1) | O(log n) |
| Sorting Courses | O(n log n) | O(n log n) | O(n) (in-order) |

## 3. Evaluation of Data Structures

Advantages and disadvantages of each data structure:

### Vector

Advantages:  
- Simple to implement.  
- Efficient for iterating and sorting.  
Disadvantages:  
- Slow for random access when searching for specific courses.

### Hash Table

Advantages:  
- Constant time complexity (O(1)) for lookups.  
Disadvantages:  
- Sorting requires additional processing (O(n log n)).

### Binary Search Tree

Advantages:  
- Automatically maintains sorted order.  
- Efficient lookups and traversals.  
Disadvantages:  
- Inefficient if unbalanced.  
- More complex to implement.

## 4. Recommendation

Recommendation: Use a Hash Table for its efficiency in course retrieval (O(1)) and overall simplicity.  
Justification: Although sorting requires additional steps, the hash table's fast access times make it the most suitable for interactive queries, which align with advisor requirements.