Vector

Operation	Total Cost	Notes
Open file and read lines	n + 1	1 for opening file + n for each line
Create Course objects	n	1 per course
Assign course data	n	courseID, courseName, prerequisites
Add to vector	n	1 per course

Total Cost: 4n + 1

Worst-case Runtime: O(n)

Hash Table

Operation	Total Cost	Notes
Open file and read lines	n + 1	1 for opening file + n for each line
Create Course objects	n	1 per course
Assign course data	n	courseID, courseName, prerequisites
Compute hash key & insert	n	1 per course (depends on collisions)

Total Cost: 4n + 1

Worst-case Runtime: O(n) if many collisions, O(1) average

Binary Search Tree

Operation	Total Cost	Notes
Open file and read lines	n + 1	1 for opening file + n for each line
Create Course objects	n	1 per course
Assign course data	n	courseID, courseName, prerequisites
Insert into BST	n^2 worst-case	O(n log n) if balanced, O(n^2) if unbalanced

Total Cost: n^2 + 3n + 1 (worst-case, unbalanced)

Worst-case Runtime: O(n^2) worst-case, O(n log n) average

Analysis of Advantages and Disadvantages

When evaluating the three data structures (vector, hash table, and binary search tree) for the ABCU Computer Science course program, I found that each has distinct advantages and disadvantages.

A vector is easy to use and simple to understand, but finding a course in it can take longer because you have to check each course one by one. Sorting the courses in order also takes extra steps, which makes it slower when working with bigger lists.

A hash table is great for quickly finding a course because it can usually find it right away. However, has tables don’t keep the courses in order, so if you want to print them alphabetically, it takes more work. They also use a bit more memory.

A binary search tree is a good balance. It lets you quickly find a course and, at the same time, keeps the courses in order so printing them alphabetically is easy. The only downside is that if the tree becomes unbalanced, searches can be slower.

Recommendation

Considering that the advisors want to see all courses in order and find course details, I would say that the binary search tree would be the best choice. It handles both tasks well, and it keeps everything organized without needing extra steps.