**Runtime analysis charts:**

**Tree:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **Line Cost** | **# Times Executes** | **Total Cost** |
| For all courses | log(n) | n | n \* log (n) |
| If the course is the same as CourseNumber | 1 | n | n |
| Print out the course info | 1 | n | n |
| For each prereq of the course | 1 | n | n |
| Print the prereq course info | 1 | n | n |
| Total Cost |  |  | 4n + log(n) |
| Runtime |  |  | O(n \*log(n)) |

**Hash table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **Line Cost** | **# Times Executes** | **Total Cost** |
| For all courses | 1 | n | n |
| If the course is the same as CourseNumber | 1 | n | n |
| Print out the course info | 1 | n | n |
| For each prereq of the course | 1 | n | n |
| Print the prereq course info | 1 | n | n |
| Total Cost |  |  | 5n |
| Runtime |  |  | O(n) |

**Vector:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **Line Cost** | **# Times Executes** | **Total Cost** |
| For all courses | 1 | n | n |
| If the course is the same as CourseNumber | 1 | n | n |
| Print out the course info | 1 | n | n |
| For each prereq of the course | 1 | n | n |
| Print the prereq course info | 1 | n | n |
| Total Cost |  |  | 5n |
| Runtime |  |  | O(n) |

**Tree:**

Some advantages of using a binary search tree data structure are that it can maintain elements in a sorted order, enabling easy printing of the course list in alphanumeric order. It also allows for searching, insertion, and deletion operations. Lastly it supports tree traversal algorithms, allowing printing of course information and prerequisites.

Some disadvantages include that trees are morememory-intensive compared to other data structures, since each course object requires more overhead for tree nodes. Insertion and deletion operations can be complex due to the balancing requirements of keeping a sorted order.

Searching for a specific course may require traversing the tree, resulting in worse time complexity.

**Hash table:**

Hash tables are moreefficient when retrieving data based on keys, in this case course numbers.They also provide constant-time average case access, allowing for quick retrieval of course information.

Hash tables can also havehash collisions, which affect overall performance. Collisions can be reduced with proper hash function design and handling.Another disadvantage is that hash tables are designed for maintaining a specific order of elements, so sorting the course list will require additional operations.

**Vector:**

Vectors aresimple when it comes to implementation and usage. They provided easy access to data, allowing for effective iteration over all the courses. Vectors also allow for dynamic resizing.

A disadvantage of this vector includes searching for a specific course. Since it requires iterating through the entire vector, resulting in a linear search time complexity.

Based on the evaluation of the advantages and disadvantages, my recommendation for the data structure would be the hash table. It provides effective case access, allows for fast retrieval of course information based on course numbers, and can handle a large number of courses with overall a great performance.