|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Line Cost** | **# Times Executes** | **Total Cost** |
| **Vector - Milestone 1** |  |  |  |
| for all courses | 1 | n | n |
| if the course is the same as courseNumber | 1 | n | n |
| for each prerequisite of the course | 1 | 1 | 1 |
| for each prerequisite of the course | 1 | n | n |
| print the prerequisite course information | 1 | n | n |
| **Total Cost** |  |  | **4n + 1** |
| **Runtime** |  |  | **O(n)** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Line Cost** | **# Times Executes** | **Total Cost** |
| **Hash Table - Milestone 2** |  |  |  |
| for all courses | 1 | n | n |
| if the course is in the hash table | 1 | n | n |
| insert into hash table | 1 | n | n |
| for each prerequisite of the course | 1 | 1 | 1 |
| print the prerequisite course information | 1 | n | n |
| **Total Cost** |  |  | **4n + 1** |
| **Runtime** |  |  | **O(n)** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Line Cost** | **# Times Executes** | **Total Cost** |
| **Binary Search Tree - Milestone 3** |  |  |  |
| for all courses | 1 | n | n |
| insert into BST | log(n) | n | n\*log(n) |
| for each prerequisite of the course | 1 | 1 | 1 |
| search the prerequisite in BST | log(n) | n | n\*log(n) |
| print the prerequisite course information | 1 | n | n |
| **Total Cost** |  |  | **n\*log(n) + 2n + 1** |
| **Runtime** |  |  | **O(n)** |

Based on the analysis, it is recommended that a Hash Table should be used for the advising program. It offers fast O(1) time complexity for search, insertion, and deletion operations, making it crucial for quickly retrieving course information and handling large datasets. Hash tables are well-suited for this purpose, as they are simple, efficient, and scalable. Although collisions may occur, these disadvantages can be minimized with a well-designed hash function and suitable collision-handling techniques. The hash table is preferred over a binary search tree (BST) due to its average-case efficiency and ease of implementation. It is the optimal data structure for the advising program, balancing speed, simplicity, and scalability.