|  |  |  |  |
| --- | --- | --- | --- |
| **Vector** | **Line Cost** | **# of times executed** | **total cost** |
| For lines in file | 1 | n | n |
| count commas in line | 1 | n | n |
| if commas less than 1 |  |  |  |
| error: not enough items | 1 | n | n |
| else |  |  |  |
| for each comma in line | 1 | n | n |
| add chars to char array within a vector | 1 | n | n |
| For each non empty char array in vector of char arrays | 1 | n | n |
| If first array in vector |  |  |  |
| Set as to courseList.at(lineNumber).courseNumber | 1 | 1 | 1 |
| if second array in vector |  |  |  |
| set as courseList.at(lineNumber).courseName | 1 | 1 | 1 |
| else |  |  |  |
| set as courseList.at(lineNumber).prerequisites.at(n) | 1 | n | n |
| **Total Cost** | 7n + 1 | | |
| **Runtime** | O(n) | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **HashTable** | **Line Cost** | **# of times executed** | **total cost** |
| For lines in file | 1 | n | n |
| count commas in line | 1 | n | n |
| if commas less than 1 |  |  |  |
| error: not enough items | 1 | n | n |
| else |  |  |  |
| for each comma in line | 1 | n | n |
| add chars to char array within a vector | 1 | n | n |
| For each non empty char array in vector of char arrays | 1 | n | n |
| currNode = new Node | 1 | n | n |
| If first array in vector |  |  |  |
| Set as currNode->courseNumber | 1 | 1 | 1 |
| if second array in vector |  |  |  |
| set as currNode->courseName | 1 | 1 | 1 |
| else |  |  |  |
| set as currNode->prerequisites.at(n) | 1 | n | n |
| hashKey = hash(courseNumber) | 1 | n | n |
| if courseList.at(hashkey) = empty |  |  |  |
| courseList.at(hashKey) = currNode | 1 | n | n |
| else |  |  |  |
| nextNode = courseList.at(hashKey) | 1 | n | n |
| while nextNode != empty |  |  |  |
| nextNode = nextNode->next | 1 | n | n |
| nextNode = currNode | 1 | n | n |
| **Total Cost** | 13n + 2 | | |
| **Runtime** | O(n) | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **BinaryTree** | **Line Cost** | **# of times executed** | **total cost** |
| For lines in file | 1 | n | n |
| count commas in line | 1 | n | n |
| if commas less than 1 |  |  |  |
| error: not enough items | 1 | n | n |
| else |  |  |  |
| for each comma in line | 1 | n | n |
| add chars to char array within a vector | 1 | n | n |
| For each non empty char array in vector of char arrays | 1 | n | n |
| currNode = new Node | 1 | n | n |
| If first array in vector |  |  |  |
| Set as currNode->courseNumber | 1 | 1 | 1 |
| if second array in vector |  |  |  |
| set as currNode->courseName | 1 | 1 | 1 |
| else |  |  |  |
| set as currNode->prerequisites.at(n) | 1 | n | n |
| insert(currNode) | 1 | n | n |
| Void insert(node) |  |  |  |
| If root == nullptr | 1 | n | n |
| Root = node | 1 | 1 | 1 |
| Else |  |  |  |
| addNode(root, node) | 1 | n | n |
| Void addNode(node, newNode) |  |  |  |
| If newNode->courseNumber < node->courseNumber | 1 | n | n |
| If node->leftChild == nullptr |  |  |  |
| Node->leftChild = newNode | 1 | n | n |
| Else |  |  |  |
| addNode(node->leftChild, newNode) | n | log(n) | log(n) |
| Else |  |  |  |
| If node->righChild == nullptr |  |  |  |
| Node->rightChild = newNode | 1 | n |  |
| Else |  |  |  |
| addNode(node->rightChild, newNode) | 1 | log(n) | log(n) |
| **Total Cost** | 13N \* 2log(n) + 3 | | |
| **Runtime** | log(n) | | |