#### **Evaluation:**

**Worst-case running time of reading the file and creating course objects**

**Vector:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Code for reading the file** | **Line Cost** | **# Times Executes** | **Total Cost** |
| string readFile(string fileName) | 1 | 1 | 1 |
| string line | 1 | 1 | 1 |
| Create object myFile from the class fstream | 1 | 1 | 1 |
| TRY | 1 | 1 | 1 |
| myFile.open(“fileName.csv”) | 1 | 1 | 1 |
| WHILE myFile is open | 1 | 1 | 1 |
| WHILE myFile is not at the end of file | 1 | 1 | 1 |
| GETLINE (myFile, line) | 1 | n | n |
| IF there are < 2 parameters on the line, THEN | 1 | n | n |
| THROW exception | 1 | 1 | 1 |
| PRINT exception msg | 1 | 1 | 1 |
| IF myFile is at the end of the file | 1 | 1 | 1 |
| CLOSE myFile | 1 | 1 | 1 |
| PRINT “file is closed” | 1 | 1 | 1 |
| CATCH myFile failures | 1 | 1 | 1 |
| PRINT “Error” message | 1 | 1 | 1 |
|  | | |  |
| **Total Cost** | | | 2n+14 |
| **Runtime** | | | O(n) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Code for creating course objects** | **Line Cost** | **# Times Executes** | **Total Cost** |
|  |  |  |  |
| Create the vector Courses | 1 | 1 | 1 |
| FOR each line in myFile | 1 | n | n |
| CREATE an object instance of the row using the constructor | 1 | n | n |
| ASSIGN each new course to an object instance variable | 1 | n | n |
| IF there are 3 or more parameters | 1 | n | n |
| FOR each prerequisite | 1 | 1 | 1 |
| IF prerequisite does not have a courseNumber THEN | 1 | 1 | 1 |
| THROW exception | 1 | 1 | 1 |
| PRINT prerequisite “NOT VALID” | 1 | 1 | 1 |
|  | | |  |
| **Total Cost** | | | 4n+5 |
| **Runtime** | | | O(n) |

**Hash Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Code for reading the file** | **Line Cost** | **# Times Executes** | **Total Cost** |
| string readFile(string fileName) | 1 | 1 | 1 |
| string line | 1 | 1 | 1 |
| Create object myFile from the class fstream | 1 | 1 | 1 |
| TRY | 1 | 1 | 1 |
| myFile.open(“fileName.csv”) | 1 | 1 | 1 |
| WHILE myFile is open | 1 | 1 | 1 |
| WHILE myFile is not at the end of file | 1 | 1 | 1 |
| GETLINE (myFile, line) | 1 | n | n |
| IF there are < 2 parameters on the line, THEN | 1 | n | n |
| THROW exception | 1 | 1 | 1 |
| PRINT exception msg | 1 | 1 | 1 |
| IF myFile is at the end of the file | 1 | 1 | 1 |
| CLOSE myFile | 1 | 1 | 1 |
| PRINT “file is closed” | 1 | 1 | 1 |
| CATCH myFile failures | 1 | 1 | 1 |
| PRINT “Error” message | 1 | 1 | 1 |
|  | | |  |
| **Total Cost** | | | 2n+14 |
| **Runtime** | | | O(n) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Code for creating course objects** | **Line Cost** | **# Times Executes** | **Total Cost** |
|  |  |  |  |
| Create the hash table Courses | 1 | 1 | 1 |
| FOR each line in myFile | 1 | n | n |
| CREATE an object instance of the row using the constructor | 1 | n | n |
| ASSIGN each new course to an object instance variable | 1 | n | n |
| IF there are 3 or more parameters | 1 | n | n |
| FOR each prerequisite | 1 | 1 | 1 |
| IF prerequisite does not have a courseNumber THEN | 1 | 1 | 1 |
| THROW exception | 1 | 1 | 1 |
| PRINT prerequisite “NOT VALID” | 1 | 1 | 1 |
|  | | |  |
| **Total Cost** | | | 4n+5 |
| **Runtime** | | | O(n) |

**Binary Search Tree:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Code for reading the file** | **Line Cost** | **# Times Executes** | **Total Cost** |
| string readFile(string fileName) | 1 | 1 | 1 |
| string line | 1 | 1 | 1 |
| Create object myFile from the class fstream | 1 | 1 | 1 |
| TRY | 1 | 1 | 1 |
| myFile.open(“fileName.csv”) | 1 | 1 | 1 |
| WHILE myFile is open | 1 | 1 | 1 |
| WHILE myFile is not at the end of file | 1 | 1 | 1 |
| GETLINE (myFile, line) | 1 | n | n |
| IF there are < 2 parameters on the line, THEN | 1 | n | n |
| THROW exception | 1 | 1 | 1 |
| PRINT exception msg | 1 | 1 | 1 |
| IF myFile is at the end of the file | 1 | 1 | 1 |
| CLOSE myFile | 1 | 1 | 1 |
| PRINT “file is closed” | 1 | 1 | 1 |
| CATCH myFile failures | 1 | 1 | 1 |
| PRINT “Error” message | 1 | 1 | 1 |
|  | | |  |
| **Total Cost** | | | 2n+14 |
| **Runtime** | | | O(n) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Code for creating course objects** | **Line Cost** | **# Times Executes** | **Total Cost** |
|  |  |  |  |
| Create the balanced binary search tree Courses | 1 | 1 | 1 |
| FOR each line in myFile | 1 | n | n |
| CREATE an object instance of the row using the constructor | 1 | n | n |
| ASSIGN each new course to an object instance variable | 1 | n | n |
| IF there are 3 or more parameters | 1 | n | n |
| FOR each prerequisite | 1 | 1 | 1 |
| IF prerequisite does not have a courseNumber THEN | 1 | 1 | 1 |
| THROW exception | 1 | 1 | 1 |
| PRINT prerequisite “NOT VALID” | 1 | 1 | 1 |
|  | | |  |
| **Total Cost** | | | 4n+5 |
| **Runtime** | | | O(n) |