**Example Runtime Analysis**

When you are ready to begin analyzing the runtime for the data structures that you have created pseudocode for, use the chart below to support your work. This example is for printing course information when using the vector data structure. As a reminder, this is the same pairing that was bolded in the pseudocode from the first part of this document.

| **Vector** | **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 information** | 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) |

| **Parser (const std::string &data, const DataType &type)** | **Line Cost** | **# Times Executes** | **Total Cost** |
| --- | --- | --- | --- |
| **If type == file** | 1 | 1 | 1 |
| **Use ifstream to open file** | 1 | 1 | 1 |
| **If file is open** | 1 | 1 | 1 |
| **While file if good** | 1 | 1 | 1 |
| **Getline from file** | 1 | 1 | 1 |
| **If line != “”** | 1 | 1 | 1 |
| **Push back line** | 1 | N | N |
| **Close file** | 1 | N | N |
| **If file size == 0** | 1 | 1 | 1 |
| **Throw error** | 1 | N | N |
| **parseFile()** | 1 | N | N |
| **Else throw error** | 1 | N | N |
| **Stream sting from data** | 1 | 1 | 1 |
| **While (getline(stream, line))** | 1 | 1 | 1 |
| **If line != “”** | 1 | 1 | 1 |
| **Push back line** | 1 | N | N |
| **If file size == 0** | 1 | 1 | 1 |
| **Throw error** | 1 | N | N |
| **parseFile()** | 1 | N | N |
| **Total Cost** | | | 8n + 11 |
| **Runtime** | | | O(n) |

| **void Parser::parseFile(void)** | **Line Cost** | **# Times Executes** | **Total Cost** |
| --- | --- | --- | --- |
| **Declare iterator** | 1 | 1 | 1 |
| **Iterate file** | 1 | 1 | 1 |
| **Declare variables** | 1 | 1 | 1 |
| **for int i=0; iterate until i is the length of the iterator** | 1 | n | n |
| **If the iterator at location i == “”** | 1 | 1 | 1 |
| **Check to see if quoted is true or false** | 1 | 1 | 1 |
| **Else if iterator at location i==’,’ && not quoted** | 1 | 1 | 1 |
| **Push row( substring iterator at start token, location i-start token)** | 1 | N | N |
| **Set start token to i+1** | 1 | N | N |
| **Push row(substring iterator at start token, length of iterator – start token)** | 1 | N | N |
| **If value is missing** | 1 | 1 | 1 |
| **Throw error** | 1 | N | N |
| **Push back row** | 1 | N | N |
| **Total Cost** | | | 6n + 7 |
| **Runtime** | | | O(n) |

| **Vector<Course> loadCourses(string csvPath)** | **Line Cost** | **# Times Executes** | **Total Cost** |
| --- | --- | --- | --- |
| **Define vector data structure to hold collection of courses** | 1 | 1 | 1 |
| **Initialize CSV parser** | 1 | 1 | 1 |
| **try** | 0 | 0 | 0 |
| **For int i, loop to read rows of CSV file** | 1 | n | n |
| **Create data structure and add to collection of courses** | 2 | n | 2n |
| **Push this course to the end** | 1 | N | N |
| **Catch error** | 1 | N | N |
| **Return course** | 1 | N | N |
| **Total Cost** | | | 6n + 2 |
| **Runtime** | | | O(n) |

| **void loadCourses(string csvPath, HashTable\* hashTable)** | **Line Cost** | **# Times Executes** | **Total Cost** |
| --- | --- | --- | --- |
| **Initialize CSV parser** | 1 | 1 | 1 |
| **try** | 0 | 0 | 0 |
| **For int i, iterate through rows of file** | 1 | n | n |
| **Create data structure and add to collection of courses** | 2 | n | 2n |
| **Push this course to the end** | 1 | N | N |
| **Catch error** | 1 | N | N |
| **Total Cost** | | | 4n + 1 |
| **Runtime** | | | O(n) |

| **void loadCourses(string csvPath, Tree\* bst)** | **Line Cost** | **# Times Executes** | **Total Cost** |
| --- | --- | --- | --- |
| **Initialize CSV parser** | 1 | 1 | 1 |
| **try** | 0 | 0 | 0 |
| **For int i, loop to read rows of CSV file** | 1 | n | n |
| **Create data structure and add to collection of courses** | 2 | n | 2n |
| **Push this course to the end** | 1 | N | N |
| **Catch error** | 1 | N | N |
| **Total Cost** | | | 4n + 1 |
| **Runtime** | | | O(n) |