| **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 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) |
| **Code for hashtable** | **Line Cost** | **# Times Executes** | **Total Cost** |
| **for all courses** | 1 | n | n |
| **if the course is the same as courseNumber** | 1 | 1 | 1 |
| **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) |

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

vector<string> split(string s, string del = " ") //splits the string at the whitespace

{

vector<string> stArray;

int begin = 0;

int end = s.find(del);

//couldn't get this to work properly.

}

void printCourse(Course course)

{

string courseNumber = course.courseNumber;

string name = course.name;

vector<string> prerequisites = course.prerequisites;

cout << "Course Number: " << courseNumber << endl;

cout << "Course Name: " << name << endl;

cout << "Prerequisites: ";

for (int i = 0; i < prerequisites.size(); i++)

{

cout << prerequisites[i] << " ";

}

}

// printing course info

void printCourseList(vector<Course> courses)

{

int n = courses.size();

for (int i = 0; i < n - 1; i++)

{

for (int j = 0; j < n - i - 1; j++)

{

if (courses[j].courseNumber > courses[j + 1].courseNumber)

{

swap(courses[j + 1], courses[j]);

}

}

}

for (int i = 0; i < n; i++)

{

printCourse(courses[i]);

}

}