# CS 300 Pseudocode Document

## Function Signatures

Below are the function signatures that you can fill in to address each of the three program requirements using each of the data structures. The pseudocode for printing course information, if a vector is the data structure, is also given to you below (depicted in bold).

// Vector pseudocode

int numPrerequisiteCourses(Vector<Course> courses, Course c) {

totalPrerequisites = prerequisites of course c

for each prerequisite p in totalPrerequisites

add prerequisites of p to totalPrerequisites

print number of totalPrerequisites

}

void printSampleSchedule(Vector<Course> courses) {

for all courses  
       print course name  
           if course has prerequisites  
               for each prerequisite  
                   print prerequisite

}

void printCourseInformation(Vector<Course> courses, String courseNumber) {

**for all courses**

**if the course is the same as courseNumber**

**print out the course information**

**for each prerequisite of the course**

**print the prerequisite course information**

}

// Hashtable pseudocode

int numPrerequisiteCourses(Hashtable<Course> courses) {

totalPrerequisites = Hashtable[c]  
   for each prerequisite p in totalPrerequisites  
       add prerequisites in Hashtable[p] to totalPrerequisites  
   print number of totalPrerequisites

}

void printSampleSchedule(Hashtable<Course> courses) {

for all key, value pair in courses  
       print key course name  
           if value has prerequisites  
               for each prerequisites  
                   print prerequisites

}

void printCourseInformation(Hashtable<Course> courses, String courseNumber) {

for all courses  
       if the course is the same as courseNumber  
           print out the course information  
           for each prerequisite of the Hashtable[course]  
               print the prerequisite course information

}

// Tree pseudocode

int numPrerequisiteCourses(Tree<Course> courses) {

totalPrerequisites = left and right child of Node c  
   for each prerequisite p in totalPrerequisites  
       add left and right Nodes of node p to totalPrerequisites  
   print number of totalPrerequisites

}

void printSampleSchedule(Tree<Course> courses) {

for all Nodes as courses  
       print course name  
           if course has left node  
                   print left node as prerequisite  
           if course has right node  
                   print right node as prerequisite

}

void printCourseInformation(Tree<Course> courses, String courseNumber) {

IF rt == null

   return

for all Nodes  
       if the course is the same as courseNumber  
           print out the node's information  
           if course has left node  
                   print left node as prerequisite course information  
           if course has right node  
                   print right node as prerequisite course information

          end Function

       else

           if course has left node  
                   goto left node  
           if course has right node  
                   goto right node

}

## 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 chart)

| **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) |

(Hash table chart)

| **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) |

(Binary tree chart)

| **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) |