Do the FLIP – Lesson Plan

College of Science – Computer Science Department

Course: CS2400 – Data Structures and Advanced Programming

Prepared and revised by: Tannaz Rezaei Damavandi

Email: tannazr@cpp.edu

Lesson: Tree and Tree traversal methods

Timeframe:

Note how long it will take learners to complete all of the activities from pre-class to post-class activities.

Pre-class: 20-25 minutes to watch two videos on:

- Tree data structure and terminologies ~ 8 minutes.
- Tress traversal methods (pre, post and in-order) ~ 15 minutes.

In-Class: 100 minutes:

- Answering questions and in class quiz on materials presented in assigned videos ~ 20-25 minutes.
- Solving more complex examples on tree concepts and different tree traversal methods ~ 20 minutes.
- Implement and program tree traversal methods using recursion. ~ 50-55 minutes.

Post-class activities:

- Implement different tree traversal methods as part of a project. (My differ between couple of hours to couple of days based on students' programming skills and proficiency)

Materials needed:

Describe items needed to complete activities:

- Two YouTube videos.
- In- class quiz (handout)
- Lecture slides
- In-class exercise (Included in lecture slides)
- Project description/instruction (posted online on BB)

Objectives:

List basic objectives tied to pre-class activities and advanced objectives tied to in-class and post-class activities.

Basic:

1- Classify linear and non-linear data structures.
2- Describe tree structure and recognize the difference between trees and graphs.

3- Describe binary trees (B-tree) and general trees using tree terminologies such as edge, node (leaf, intermediate and root), parent, child, siblings, height and depth.
4- Traverse a B-tree in four traversal methods (pre-order, post-order, in-order and level-order)

Advanced:

1- Perform general tree traversal (pre-order, post-order, in-order and level order)
2- Design B-tree and B-tree traversal methods' algorithms and pseudocodes.
3- Construct B-tree and implement B-tree traversal methods in java programming in java programming language.

Background ("Situation"):

Note the typical composition of learners in the class, how this lesson fits into course design/schedules, prerequisite knowledge requires, and typical challenges that learners face with this content area.

Typical composition of class is mainly comprised of computer science major students (sophomore to senior levels); however, rarely other major students who are willing to learn the bases of programming may enroll in this class.

Tree is the first lesson of nonlinear or hierarchical data structures; it fits between nonlinear (Abstract Data Type) ADTs (arrays, linked lists, stacks and queue) and graphs. In semester system all tree lessons will be taught toward the end of semester (weeks 11-13).

Students need to know how to use/call all linear data structures in java. A typical challenge that students may encounter is implementing this ADT on their own as opposed to linear data structures that are already built-in. The other challenge that students may face is implementing tree traversal methods. Students need to have a solid knowledge on java programming and be good at object oriented concepts such as object, class, inheritance and …

Introduction to Lesson:

Describe the purpose of this content area for learners and an overview of the activities and resources for the flipped lesson.

We often must categorize data into groups and subgroups that cannot be done with linear data structures. Hence, we need to define a new abstract Data Type (ADT) that can contain data items appeared at various levels. A tree is a hierarchical (no-linear) ADT that is widely used to show data items with ancestors and descendants. Family tree, organizational hierarchy and file directories are some examples for tree data structure. Following figure shows a portion of the administrative structure of typical university.

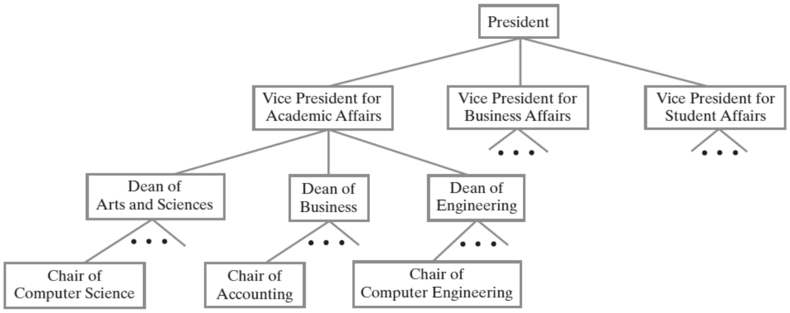
President
Vice President for Academic Affairs
Vice President for Business Affairs
Vice President for Student Affairs
Dean of Arts and Sciences
Dean of Business
Dean of Engineering
Chair of Computer Science
Chair of Accounting
Chair of Computer Engineering

Figure 1- Portion of Administrative Structure of a Typical University [1]

In this lesson, you are responsible to watch two YouTube videos. In these videos you will learn about trees, tree terminologies and tree traversal methods. After watching these two videos you should be able to define trees, describe tree terminologies, and traverse a binary tree in one of four ways (Pre-order, in-order, post-order and level-order).

Class starts with a short Q&A session following by a quiz on pre-class materials. All students are encouraged to answer their classmates' questions. Perfect answers will receive extra credit points.

The main focus of class will be on constructing a B-tree and implementing B-tree traversal methods in java. During class time you need to have your computer and work in a group to do the programming in-class assignments.

As a post-class activity, you will need to apply what you have learned in this lesson (constructing B-tree and B-tree methods) as part of a project which worth 10% of your final grade.

Procedure:

Pre-class individual Space Activities and Resources:

Outline major steps for preparatory activates, ensuring steps lead to mastery of basic objectives you have noted above. Note resources required for learner preparation.

Steps	Purpose	Estimated Time	Learning Objective
Step 1: **Watch** Introduction to trees	To learn tree and tree terminologies.	~16 minutes	Basic objectives: #1-3
Step 2: Watch Tree Traversal Methods	To learn different methods of tree traversal.	~ 8 minutes	Basic objectives: #4

In-Class Group Space Activities and Resources:

Outline steps for in-class activities, ensuring that activities lead to mastery of advanced learning objectives you have noted above. Note any resources needed/developed for activities.

Steps	Purpose	Estimated Time	Learning Objective
Step 1: **In class Q&A on pre-class assignment**	To clarify any ambiguity on pre-class assignments. Students can ask questions and their classmates are encouraged to answer. Perfect answers will receive extra credit points. Instructor will elaborate on students' answers if they need any.	~ 5-10 minutes	Basic objectives: #1-3
Step 2: **In Class quiz**	To evaluate students basic knowledge on pre-class assignments	~ 15 minutes	Basic objectives: # 1-3
Step 3: **Complex problem solving**	To learn traversing B-tress and general trees. Handout will be provided either in	~ 20 minutes	Advanced objectives : #1

	hardcopy or electronic format.		
Step 4: Constructing B-tree and B-tree traversal methods	To learn how to program and construct a B-tree and its traversal methods in java. In this step, students participate in groups of 3-4 and do programming tasks under instructor guidance and supervision.	~ 50-55 minutes	Advanced objectives : #2-3

Post-Class individual Space Activities and Resources.

Outline steps for post-class activities, ensuring that activities lead to mastery of advanced learning objectives you have noted above. Also note resources learners will need to complete class activities.

Steps	Purpose	Estimated Time	Learning Objective
Step 1: Assignment	More practice on materials presented in pre-class and in-class activities.	1-8 hours. It depends on students' learning speed and programming skills.	Advanced objectives:#1-3
Step 2: Implementation	Evaluate students understating on presented materials by doing a hands-on project. They need to construct B-tree and implement its traversal methods as part of a project which worth 10% of their final grade.	Couple of hours to couple of days. Depends on students' programming proficiency.	Advanced objectives: #1-3

Resources that students may need:

- Pre-class and in-class materials.
- Course Text Book.
- Online resources such as: YouTube videos, stackoverflow.com and any online resources that appeal to students.
- Taking advantage of instructor's office hours.
- Department tutoring center and resources.

Evaluation:

Analysis :

Note what you think will work, and challenges you anticipate.

I think all provided materials are sufficient enough to make students master this lesson. Hence, if students follow the instructions, they will grasp the concept and can do all assignments. My major concern is encouraging students to do pre-class activities. In order to do so, I use the most useful pre-class videos.

They are sufficient and concise; they contain all points and concepts that students need to know to learn this lesson. In addition, at the beginning of the class and after a short Q&A session, I will quiz students. During Q&A all students have a chance to answer their classmates' questions. Perfect answers will receive extra credit points. The in-class quiz is a good tool to encourage students not only to study pre-class materials but also to attend the class. In addition, I can use the score as an indicator to evaluate learning outcomes.

Another major challenge is when some students don't have solid required programing skills and may fall behind as we progress. I think assigning team projects in which students can collaborate and learn from each other is a good solution. In addition, I encourage them to take advantage of my office hours and school tutoring centers and resources.

Connections to future lessons:

Note how this lesson plan connects to your next topics in the course.

This lesson plan is the foundation for the next lessons on tree and its implementation. Next lesson will focus on B-tree representation and B-node class implementation.

Reference:

[1] Henry, T. (2015). Data Structures and Abstractions with Java. Pearson Education Limited.