Thomas Keller

Menu():

Int case

Switch(case):

Case 1:(load)

Call upon parser to load items into a binary search tree

Case 2:(print list)

Print list in alphanumeric order

Case 3:(print individual)

Use search algorithm to grab desired target, if not print not found

Case 4:(exit)

Print goodbye to console and exit program.

BinarySearchTree InOrderPrint(Node n):

If n is null

Return

InOrder(n.left)

PrintCourse( n.course)

InOrder(n.right)

Vector quickSort(vector v, int begin, int end):

Int mid = 0

If begin is less than end

Find the middle value using partition then call quicksort on both

partition(vector v, int begin, int end):

set low and high equal to begin and end

pick the middle element as pivot point

while not done

keep incrementing low index while bids[low] < bids[pivot]

keep decrementing high index while bids[pivot] < bids[high]

else swap the low and high bids (built in vector method)

move low and high closer ++low, --high

If there are zero or one elements remains, all bids are partitioned. Return high

Print vector()

Print in order since vector will be sorted.

Evaluation

The program goes line by line through the csv, making sure there are at least enough values to make a course object. The parser will input the information into first the course information, then the course prerequisites. The parser simply creates a new course object with the first input as the base class, and uses methods to add prerequisites. This should run in O(n) time because each line is visited once, then parsed into the course class. The creation and input of data into a course class should take O(1) time as well. The advantages of binary search trees is that the insertion is already sorted, so classes can be easily gotten to and printed in order. The advantages of a vector is that they can be quickly sorted and hash tables have a very low search time. The problem with binary search trees is that the search times can vary and the tree will not be balanced if the values are already in order. Vectors are not very malleable and can lead to long traversal times for searches. Hash tables are very hard to sort, and I could not think of a very good way to print the contents in order.