Concepción Ardon

CS300

04/08/2022

Pseudocode 1

ABC’s Advising Program

Start program

OPENFILE

READ file data

INIT number Prerequisite Courses

Total prerequisites are equal to prerequisite of course c

FOR each prerequisite in the total prerequisite

add the prerequisite to the total prerequisites

DISPLAY number of total prerequisites

ENDFOR

FOR all courses

IF course is the same as course number

DISPLAY course number, title, and prerequisites

ENDIF

ENDFOR

FOR each prerequisite of the course

DISPLAY prerequisite course information

ENDFOR

END Program

ABC’s Advising Program Pseudocode menu

Start the program

Display the menu

GET user to input

IF option is equal to 1 THEN

Load the file data into the data structure

Display the data structure

Return to menu

ELSEIF option is equal to 2 THEN

FOR each course in the list

Sort the list in alphanumeric order

Display course list

ENFOR

Return to menu

ELSEIF option is equal to 3 THEN

Display course title and prerequisites

Return to menu

ELSEIF option is equal to 9 THEN

Display a goodbye message

END program

ELSE

Display error message for invalid input

Return to Menu

ENDIF

Runtime Analysis

| Code | Line Cost | # Time executes | Total Cost |
| --- | --- | --- | --- |
| For each prerequisite in the total prerequisite | 1 | n | n |
| add the prerequisite to the total prerequisite | 1 | 1 | n |
| Print the number of total prerequisites | 1 | n | n |
| For all courses | 1 | n | n |
| if the course is the same as the course number | 1 | 1 | n |
| Print the course number, title, and prerequisites | 1 | 1 | 1 |
| For each prerequisite of the course | 1 | n | n |
| Print prerequisite course information | 1 | n | n |
|  |  | Total Cost | 5n+1 |
|  |  | Runtime | O(n) |

Analysis report

Vector data structure's advantages include its ability to represent topology well, its ability to be easily adapted to scaling changes, and its ability to be used with attributed data. Vector data structures have the disadvantage that they are complex, difficult to overlay, difficult with image processing, and require constant updating. The advantages of hash table data structure are synchronization, simple with no extra structures needed, and very space-efficient. In addition, hash collisions with hash tables are practically unavoidable, hash tables become inefficient, and hash tables do not support null values. The advantages of trees are that they keep data sorted and make finding data extremely fast, they make tree growth make searching less and less slow, they allow you to add and remove data from trees very quickly just by changing the references to the child node nodes like in linked lists, and they are relatively efficient in terms of memory since they only allocate enough for storing the values in the nodes and the references between them. The disadvantages of tree data structures are that trees require memory to keep their references to children, that they must be balanced to be efficient for search, and that memory usage is costly. Trees are also slower than hash tables and vector data structures.

I recommend that I use the tree data structure for the program. Because the tree data structure stores the information that naturally forms into a hierarchy. By arranging the keys into a tree, we can search for a given key in a reasonable amount of time. I can insert or delete keys in moderate time. A tree that allows fast search, insert, delete on a sorted data. It also allows finding the closest item. Like linked lists, Pointer implementation of trees don’t have an upper limit on number of nodes as nodes are linked using pointers. Furthermore, tree data structure has more of an advantage in my view and should be used to code the program.