This is a graded discussion: 10 points possible

due -

16

Week 6 - Linear vs. Tree Data Structures

What is the advantage of utilizing trees as a data structure? Describe a scenario where you may use either a linear data structure or a tree and show which would be more effective and why?

Answer the prompt and respond to at least two of your peers' posts. You must make an initial post before you are able to view the posts of your peers. To view the discussion board rubric, click the gear icon in the upper right corner and select "Show Rubric."

Search entries or author | Unread | ✓ Subscribed

Reply

○

(https:/

Tyler Porter
(https://maryville.instructure.com/courses/43640/users/66672)

Tuesday

The advantage of using a tree as a data structure is that there are times when you need to put information into a hierarchy. It makes me think of how our modules are set up for the classes. You can't have week 2 until you complete week 1, or even something as simple as files saved on your computer. The tree can be used to organize the keys that we can then search any key in a moderate amount of time. This is going to be faster than a linked list but not as fast as an array. Also, we can add or remove keys in a moderate amount of time which is faster than an array but slower than an unordered linked list. It all really depends on the amount of time you are needing. A tree is going to take less time than a linear data structure.

Reply

○

(http

Christopher Jackson
(https://maryville.instructure.com/courses/43640/users/66606)

Wednesday

Hey Tyler,

Great example with the modules! I didn't notice that we were using a hierarchal structure this entire course. It makes you wonder what other real world things that we interact with on a daily basis use a tree data structure and some of the other concepts we are/will continue to learn throughout this program. Thanks for sharing!

Reply

○

(https:/

Tyrone Davidson (https://maryville.instructure.com/courses/43640/users/66674)

Wednesday

One advantage of utilizing trees as a data structure is it helps manage the creation and organization of files and folders. A scenario where I would have to use a tree data structure is if I were creating a find friend or contact function on a device. The function that lets you sort through your contacts by the letters you type into it. For example, if I am searching for a “John” in my list, when I type in “J-o” every name that starts with those letters begin to appear as a suggestion. This makes it easier to locate contacts opposed to typing someone’s full name. A linear data structure, would accommodate by the order in which names or contacts were added to my list. A tree would sort through the list by reference. Therefore, using a tree data structure would be more effective.

Reply

○

(http

Eli Hadam Zucker (https://maryville.instructure.com/courses/43640/users/33096)

Wednesday

Hey Tyrone,

Thanks for your post today.

The way that you explained the program trying to search for the names is a perfect example of a tree data structure at work. Another example that I can think of is a file path for a computer as you type the name of the file in the search bar. The search program has to search through the entire disk tree to try to find the data string that matches the query. Thanks for sharing!

Reply

○

(http

[Tyler Porter](https://maryville.instructure.com/courses/43640/users/66672)
(https://maryville.instructure.com/courses/43640/users/66672)

Yesterday

Tyrone,

I hadn’t thought about the contacts in our phones being a data tree. You mentioned that a linear structure of this would be based on when they were added to your list. However, it could also be a hierarchy based on the alphabet.

Reply

○

(https:/

[Patrice Gardner](https://maryville.instructure.com/courses/43640/users/68137)
(https://maryville.instructure.com/courses/43640/users/68137)

Wednesday

Tree data structures is a great function to organize data in general. In programming, there are multiple objects and commands that are linked to each other, forming some type of relationship. Because it also establishes hierarchy through nodes, and order can be followed for better understanding and applications of algorithms based on priority. Overall, tree data structures can exceed performance of how well a program performs based on the advantages of the structure, itself. For example, a tree data structure can be used as a filing system. The structure allows for efficiency, especially when considering tree traversals. The structure can help navigate to particular information of stored data, removing unnecessary steps. Therefore, in the filing system, one can travel down the left or right side of a root node looking for a particular file and continue down each section/subsection of the child to locate the specific data, using the preorder pattern.

Linear Data Structure is also a great organizational tool but limited to operations such as where new elements can be added. It operates off of sequence, so new data can be entered in the beginning or end and sorted but isn’t linked to a specific root for relation. If we take my previous example of the file system, a tree data structure would be more useful. If we had a folder in the file system for “case workers”, leading to a section for “full-time” and “part-time” , along with branched off sections for “yearly cases” of each, then we can showcase this as a organizational tree, still reverting back to the root. However, in a linear data structure, branched off sections or elements cannot stem from a sequence.

Screen Shot 2020-04-15 at 7.38.39 PM.png (https://maryville.instructure.com/files/7722641/download?download_frd=1&verifier=H0f7vzZSX08gBb2CGGDzFTbl0HaGpX8HuxjfzA6F)

Reply

○

(http

Christopher Jackson (https://maryville.instructure.com/courses/43640/users/66606)

Wednesday

Hey Patrice,

I agree with your points of how a tree data structure shows major advantages with hierarchal relationships through data nodes and speed. Also, I agree with your thoughts on the limitations of the linear data structures. After learning about how effective a tree data structure is it showed me that there are best solutions for certain problems. Knowing about all of the advantages and disadvantages can help a programmer know the best time to use a certain tool. Your graph is awesome too. Thanks for sharing!

Reply

○

(https:/

Christopher Jackson (https://maryville.instructure.com/courses/43640/users/66606)

Wednesday

Some of the main advantages of utilizing trees as data structures are the efficiency, hierarchal representation, and structural properties. The efficiency of the operations to search, insert, and delete data in a tree data structure have a better time complexity compare to other data structures. The time complexity of the search, insert, and delete operations are all O (n) in tree data structures. Another advantage of using a tree data structure is the hierarchal representation of data that the structure has. This can be useful for solutions that need to show a hierarchal representation between different nodes of data in their data set. Lastly, the structural properties are another advantage of utilizing a tree data structure because of the ability to show the relationship between items in your data. For example, showing how one sub set of items may be related to each other while another set may be independent of another sub set.

A scenario where you may use a tree data structure is creating a family tree. A family tree is hierarchal and shows relationships between all of its family members. For example, there will be great grandparents, who have offspring and their offspring will have more offspring and so on - creating a family hierarchy. A tree data structure in this example is effective because it will show the parents and all of their children as well as sub sets of immediate family members. A linear data structure in the family tree example would not be effective at all because it does not show any of the useful information on who is the parent, who is the child, where the immediate families are, and who is in a sub tree with whom.

Reply (1 like)

○

(http

Wyatt (https://maryville.instructure.com/courses/43640/users/66677)

Yesterday

Hi Christoper,

I really like your point about trees showing relationships and the example of a family tree really helps drive the underlying principle home.

Thanks, Wyatt

Reply

○

(http

Tyler Porter (https://maryville.instructure.com/courses/43640/users/66672)

Yesterday

Christopher,

A family tree is an excellent example of a tree data structure. Especially how it shows the relationships between each of the members. Anytime you see a family tree you can easily see where people fall. It would definitely take more time to determine relationships if the tree was in a linear structure.

Reply

○

(https:/

Eli Hadam Zucker (https://maryville.instructure.com/courses/43640/users/33096)

Wednesday

To determine the advantage of one data structure over another; one must generally conduct a survey of needs for their organization. This survey of needs will allow for the administrator to know the overall composition of the data. If the data is hierarchical (like a university or college administration or file system on a computer may be); then a tree data structure may make more sense compared to a linear data structure.

If you have a file path for a computer program, it would make more sense to code whatever program you're working on to be able to be called through that file path; which would allow for the program to be called more quickly than if it was a Linear data structure. If you needed to code something with a stack, a linear structure would make more sense because it's dealt with in a single run.

Reply

○

(https:/

Samuel Rose

(https://maryville.instructure.com/courses/43640/users/66999)

Wednesday

The advantage of utilizing trees is that they are very efficient for memory management. I great scenario for when a tree might be used is when you have thousands of files or data nodes that contain a particular piece of information. Each piece of information is tied to a particular individual as well. If we have a system that had to locate specific information given a specific individual and that person is within those thousands of files a linear data structure sort becomes inefficient. I recently had a similar scenario and given the volume of files I was sorting through I capped my RAM usage due to particulars of the program. Implementing a tree structure for the exact same cases minimized the memory usage and kept my computer from crashing.

Reply

○

(https:/

Kristin Sebert

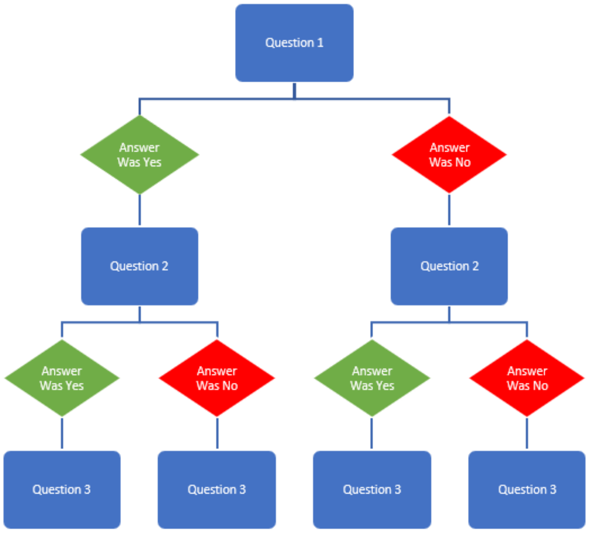
(https://maryville.instructure.com/courses/43640/users/67340)

Wednesday

Trees seem to provide an advantage as a data structure for storing anything that is already in a hierarchical relationship where it is important to retain the relationship among the data elements. As the text notes, a tree data structure is very effective for storing the directory information for a file system and for storing data for a family tree.

For an alternative example, imagine that we are storing the results of a series of sequential guesses:

Twenty Questions Decision Tree



Although we could easily store the questions and answers in a linear data structure, we couldn't easily represent the way that the guesses diverged based on the answers provided.

Reply

○

(http

[Wyatt](https://maryville.instructure.com/courses/43640/users/66677)

(https://maryville.instructure.com/courses/43640/users/66677)

Yesterday

Hi Kristen,

I like the picture of a decision flow chart you posted. Between Samuels and your post, it got me thinking about situations that tree and linear data structures work together.

The example I came up with was using your decision tree, some code could be instructed to work through and process a linear data structure.

Thanks, Wyatt

Reply

○

(https:/

[Nathan Tanthavong](https://maryville.instructure.com/courses/43640/users/31041)

(https://maryville.instructure.com/courses/43640/users/31041)

Thursday

The major advantage of a tree data structure is that it is nonlinear. Linear data structures can take a long amount of time to search for data since the time it takes to find what you are looking for depends on how big the data structures is. This is because you have to traverse the data structure in a linear manner to find the data you want. Because trees are nonlinear, you can the hierarchical system to find data quicker.

A good example of a tree implementation would be if you wanted to create a hierarchy for your company and all its employees. It would be split up into departments and sub departments. A tree would be better for this because if you were looking for someone in the IT department you can ignore all of the other departments, whereas if it were a linear structure. You might have to go through the account department, for example, in order to get to IT. Therefore a tree would have a quicker running time than a linear data structure.

Reply

○

(https:/

Wyatt
(https://maryville.instructure.com/courses/43640/users/66677)
Thursday

Hello All,

Linear trees allow us to model data and their relations in new and more efficient ways. For example, you technically could model your family tree in a linear data structure such as a Stack, it makes much more sense to model it in a Tree structure. This ways to our mental model better, as well as would make querying the data much more efficient.

Here is another good article on Trees I found while studying: **https://www.freecodecamp.org/news/all-you-need-to-know-about-tree-data-structures-bceacb85490c/** (https://www.freecodecamp.org/news/all-you-need-to-know-about-tree-data-structures-bceacb85490c/)

Cheers, Wyatt

Reply