

Tree node structure

Advantages

The ability with pointers and encapsulated data is really great to have as we are able to use functions that correctly pair with the algorithm of choice so it's less difficult and more efficient to implement specific tasks such as search, add, and remove.

Disadvantages

It uses a lot of memory due to the amount of pointers used, in addition to that list traversals and reverse traversals make it more difficult to access individual nodes

Binary search tree algorithm

Advantages

For large data sets, we are able to keep lookups in logarithmic time so insert() delete() lookup() are all to $O(\log N)$

It's possible to do range queries between two keys

We are able to speed up chunks of keys that are considered free blocks so we do not have to waste extra time and computing power while iterating through them

Disadvantages

A disadvantage is that in the case the tree is not balanced it essentially is deduced to a linear search on an array and we lose the cost of operations as it may not be logarithmic

- Brainstorm other possible applications for the pair. How could the pair be used to solve problems relating to your own personal or professional interests? Illustrate your response with specific details.

Any instance in which there is a problem that requires comparing two types of variables in a greater or less than fashion, for example if I had wanted to choose a movie to watch basing the decision of whether or not I would be more or less likely to watch the movie based off of a set of parameters distinguishing type, in addition to comparing the ratings of each movie I could use a

binary search tree. In addition to this, every time I watched a movie, and gave it my own rating, I could teach the program to grow and learn, tweaking its parameters the more I used it, really refining the decision making process of the software implemented within the recommendation system.