

Quiz #4 Answers

1. What does it mean to augment a data structure? Give an example. What is the primary consideration, after correctness, which you have to make when you augment a data structure?

Augmenting a data structure is adding some extra piece of data to the structure that allows us to perform some extra operation with that structure. An example is adding sizes to a binary tree to make an “order-statistic” tree. An interval tree was also an example where we added the maximum interval under each node. If we want to augment a data structure the one thing we have to be absolutely certain of is that we can maintain the new information without altering the O speed of the methods. For example, if we augment a tree, the add, remove, and search algorithms all have to remain $O(\log n)$. This implies that we can't want the whole tree to keep things updated.

2. Compare and contrast the two methods of spatial trees that we have discussed. (KD-tree vs. quad/oct-tree)

Both of these types of trees are spatial trees. So we are ordering data elements based upon several values. The quad/oct-tree approach breaks up the space in a very regular way forming 2^d equal sized subcells, where d is the dimensionality of the space. The KD-tree only breaks the set up into two pieces at each step. In that sense it is like a binary tree. However, the breaks can occur perpendicular to different axes. Each internal node gives an axis that the cut is made perpendicular to and the location of the cut. In this sense, the KD-tree is more flexible because we adjust where cuts are made, but it also means that we have to do extra work to try to figure out where those cuts should be made.

Extra Credit: What is the main reason a system would use a B-tree?

If it is too big to fit in memory and needs to sit out on disk then a B-tree is optimal.