Kishan Patel

CSC 212 Research Project

Apr 16, 2024

Range Tree

A range tree is an ordered tree data that allows efficiently reporta all points within a given range and is typically used in two or higher dimensions. Range trees were first introduced by Jon Louis Bentley in 1979 and are an alternative to the k-d tree with range trees offering faster query times but worse storage. That being said, range trees are mostly used for very efficiently solving multidimensional range queries.

A Range tree involving 1 dimension is essentially a balanced binary search tree in which at any node, the left sub-tree will have the values less than or equal to the node value and in the right tree the value is greater than the node value. When extending to higher dimensions, range trees are built recursively by constructing a balanced binary search tree on the first coordinate of the points, and after that, for each vertex v in this tree, building a (d−1)-dimensional range tree on the points contained in the subtree of v, resulting in a time complexity of O(n log^d n). In order to solve a range query, the tree is recursively traversed and reports the maximum value of the nodes that is within the interval of the query. This range query is very efficient with having the time complexity O(log n) where n is the number of data points.

Range trees are most effective at performing range queries in multidimensional data sets. They efficiently and quickly return a set of values within a specified range and are mostly ideal in applications such as spatial databases and computational geometry. Since range trees are constructed from a balanced binary search tree, this ensures efficient search, insertion, and deletion operations as well with the time complexity being O (log n).

The main disadvantages of the range tree are that they have a limited area of uses and that it requires lots of memory. While range trees are efficient at range queries, they may not be the best choice for other types of queries or operations such as nearest neighbor or exact match queries. Since range trees are effective at retrieving points within a certain range in a multidimensional space, therefore, other types of queries involving different types of data may be more suitable. Range trees also require a lot of storage because of the structure of the tree, especially in higher dimensions or larger datasets. Since range trees are balanced trees, additional storage is needed to perform rotations or rebalancing operations when necessary.

The range tree was chosen for this project because of its effectiveness in answering range queries. The project requires retrieving classes within a specified range based on the capacity of the class. The range query allows the program to account for what classes are open and return classes with more capacity to be filled up first. The implementation of the range tree seemed the easiest to comprehend when trying to find what classes should be filled in first due to the class itself and the capacity of it.