Augusto Cohn

COP4720

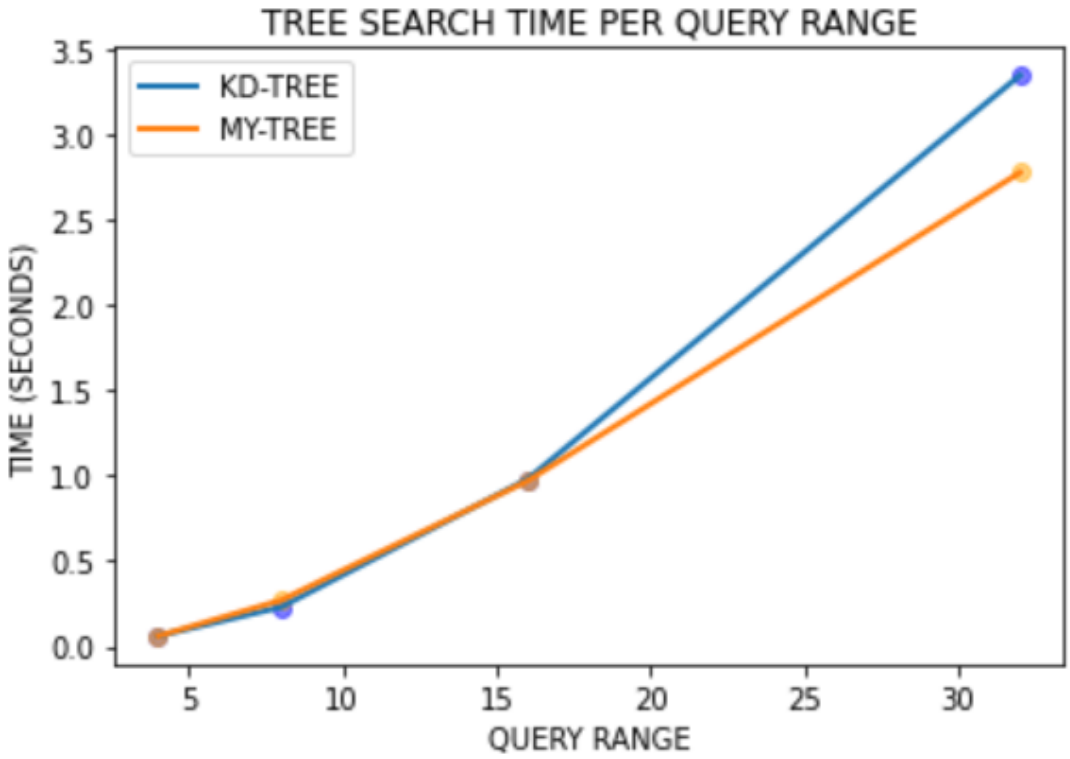
Programming Assignment Report

BREIF:

My-tree is built by splitting data on the dimension that has the largest range. Every time a non-leaf node is generated, the incoming data is loaded into separate vectors by dimension. The range of each dimension is calculated and the dimension that has the largest range is what the incoming data is split on. This means that each level can have different dimensions in which the data is split on. The purpose of this was an effort to minimize the depth of the tree by finding a more optimal dimension to split on (the mean of the largest range was the splitting point). If the tree gets shallower and more balanced, there are less nodes needed to traverse to the leaves, speeding up the search times.

DISCUSSION:

```
X values:  [4, 8, 16, 32]
Average KD-TREE search times:  [0.06, 0.23, 0.98, 3.35]
Average MY-TREE search times: [0.06, 0.27, 0.97, 2.78]
```



For the 4 queries searched, above is a graph generated with Matplotlib.pyplot. Each query was ran 3 times on a fresh run of rangeQ and the average value was used to generate this graph. The timer for the search was starting as soon as the building of the tree was completed and queries started executing. The timer stops after each query is finished executing. **Something important to note:** Each record that was found to be relevant to a query was first added to a buffer vector so that the results can be sorted. This is the case for both MY-TREE and KD-TREE. This was done to make comparisons in the outputs viable using diff. For smaller query ranges, MY-TREE and KD-TREE search times are very similar. Differences in search times can be attributed to randomness. However, as the query range increases, MY-TREE appears to have a small advantage with the standard projDB database which contains 2,000,000 normally distributed records ranging from 0-1024 on the x and y dimensions. I anticipate that with more skewed, larger or higher dimensionality databases, the advantage of using MY-TREE will become greater. MY-TREE build is slightly computationally more expensive as vectors are being build along each dimension every time a non-leaf node is created in order to find the minimum and maximum values.