Assignment 3

Note:

- Any form of copying (even in code), or allowing someone to copy, will not be tolerated. Plagiarism will result in an F grade.
- You can do this assignment in groups of 3
- Due date: 11:55 PM on 16th April.

---

1. Implement a KD-tree for any arbitrary dimension up to 20. Obviously, you should ensure your implementation is as fast as possible. You may also assume that each dimension takes only non-negative values, i.e., no dimension of a point contains a value less than 0.
   a. Generate uniformly distributed 100,000 points where each dimension takes a double value between 0 and 1. Next generate random query sets of 100 points each for dimension values 2, 3, 5, 10, 15, 20. Perform 20-NN query for each point in the query set. Plot the average running time per query point (y axis) against dimension (x-axis). In addition, also plot the running time of sequential scan against the dimension. You should implement the k-NN algorithm using best first search algorithm as explained in class. The sequential scan should be implemented using max-heap of size k so that the complexity is $O(nlogk)$. Use L2 distance. [10 points]
   b. For the same query sets as (a), perform 100-NN query. Compute the average distance of a query point to the second closest point and the 100th closest point. Plot the ratio of these two distances (numerator 2nd closest point distance and denominator 100th closest point distance) (y-axis) against dimension (x-axis). [10 points]
   c. Explain the reasons behind the trends you see in the plots for (a) and (b). Keep your write-up precise and short (at most 200 words). [10 points]
   d. Efficiency Competition: We will have a competition on the running time of the k-NN query among all submitted implementations of the kd-tree. You are NOT allowed to change the KD-tree structure or use sequential scan to answer K-NN queries. Refer to the following link for the detailed instructions and to pull the starter code: [30 points] https://github.com/abhi19gupta/KdTree-StarterCode
      i. The grading criteria for this competition is as follows
         1. $Score = \left(\frac{fastest\ time}{your\ time}\right) \times 30$
      ii. We will run multiple queries and your final score would be the average score of them all. Only 100% accurate answer sets would be counted.

Submission format:

Put all your code, run.sh and writeups in a directory named <entry no.1>_<entry no.2>_<entry no.3> and zip the folder. The zip file should be named as <entry no.1>_<entry no.2>_<entry no.3>.zip. (Please note: Use your entry numbers of the format 2014CSXXXXX and not the Kerberos Ids)