HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

Fall 2017

T-301-REIR, Reiknirit

S3: Kd-Trees

Halli og Laddi
kt. 123456-7890
Group 1

September 14, 2017

TA: Eiríkur Fjalar

Directions on performing the assignment are showed here in italics (like this). These should not be included in the report you submit.

1 Implementation

Node

Describe the Node data type you used to implement the 2d-tree data structure.

Range Search

Describe your method for range search in a kd-tree.

Nearest Neighbor Search

Describe your method for nearest neighbor search in a kd-tree.

2 Analysis

Memory

Give the total memory usage in bytes (using tilde notation and the standard 64-bit memory cost model) of your 2d-tree data structure as a function of the number of points N. Justify your answer below. Include the memory for all referenced objects (deep memory), including memory for the nodes, points, and rectangles.

bytes per Point2D: 32 bytes
bytes per RectHV:
bytes per KdTree of N points (using tilde notation): $\sim$
(include the memory for any referenced `Node`, `Point2D` and `RectHV` objects)

Running Time

Give the expected running time in seconds (using tilde notation) to build a 2d-tree on N random points in the unit square. Use empirical evidence by creating a table of different values of N and the timing results. (Do not count the time to generate the N points or to read them in from standard input.)

Nearest Neighbor

How many nearest neighbor calculations can your brute-force implementation perform per second for input100K.txt (100,000 points) and input1M.txt (1 million points), where the query points are random points in the unit square? Explain how you determined the operations per second. (Do not count the time to read in the points or to build the 2d-tree.) Repeat the question but with the 2d-tree implementation.

Table 1: !Insert caption!

calls to `nearest()` per second

	brute force	*2d-tree*
`input100K.txt`		
`input1M.txt`		

3 About This Solution

Have you taken (part of) this course before:

Hours to complete assignment (optional):

3.1 Known Bugs / Limitations.

Known bugs / limitations. For example, if your program prints out different representations of the same line segment when there are 5 or more points on a line segment, indicate that here.

3.2 Help Received

Describe whatever help (if any) that you received. Don't include readings, lectures, and classes, but do include any help from people (including course staff, lab TAs, classmates, and friends) and attribute them by name.

3.3 Problem Encountered

Describe any serious problems you encountered.

3.4 Comments

List any other comments here. Feel free to provide any feedback on how much you learned from doing the assignment, and whether you enjoyed doing it.