Language:

C++

the environment:

Mac

type of compiler: g++

Configured with: --prefix=/Library/Developer/CommandLineTools/usr --with-gxx-include-dir=/usr/include/c++/4.2.1

Apple LLVM version 7.3.0 (clang-703.0.31)

Target: x86\_64-apple-darwin15.5.0

Thread model: posix

Name:

周昱岑（Yu-Tsern Jou）

Student ID No:

B01901082

Phone number:

0927-991-817

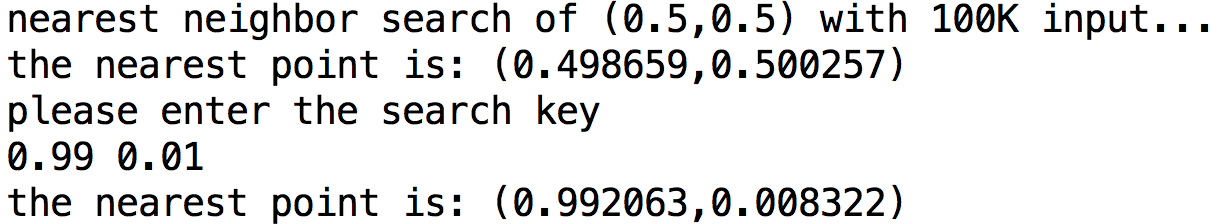
Execute procedure:

Enter “g++ main.cpp KDTree.cpp RBTree.cpp” in the command line

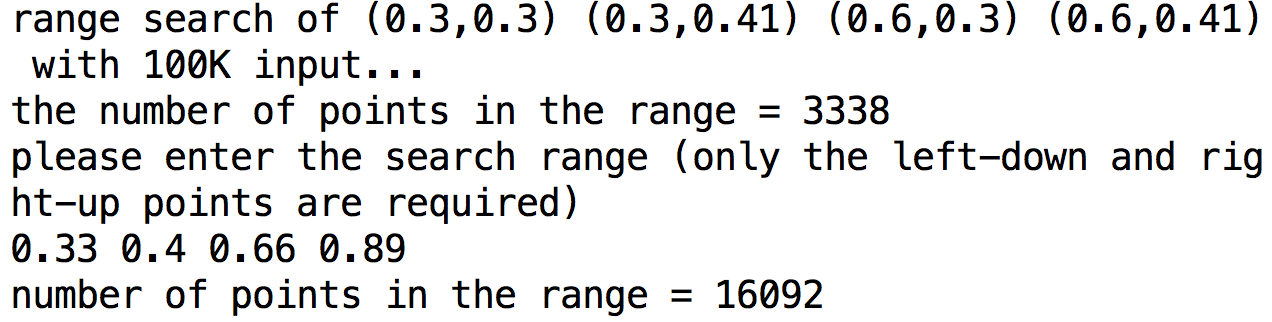
Enter “./a.out” in the command line

Follow the instructions key in search keys or search ranges during executing (2 or 4 floating point numbers in (0,1). In the case of range search, only the left-down and right-up positions are required.)

Example are as follow:



This picture shows the case of nearest neighbor search



This picture shows the case of range search

Source code:

There are five parts of source code: main.cpp, KDTree.cpp, RBTree.cpp, KDTree.h, RBTree.h. Each of the .h file specified the class members, while the KDTree.cpp and RBTree.cpp implemented the declared functions. At last, they are all utilized in the main.cpp to meet the needs of this homework assignment. Specifically, in case there is a need for checking ways of implementation, there are divided sections in main.cpp corresponding to each of the problem.

Analysis:

2.(a) time consumed when building a 2D tree.

|  |  |  |
| --- | --- | --- |
| Number of node | Time consumed | Log(timed consumed) |
| 1000 | 0 | X |
| 10000 | 0 | X |
| 20000 | 0 | X |
| 40000 | 0.01 | -2 |
| 80000 | 0.02 | -1.698970004 |
| 100000 | 0.02 | -1.698970004 |
| 200000 | 0.07 | -1.15490196 |
| 400000 | 0.19 | -0.721246399 |
| 800000 | 0.5 | -0.301029996 |
| 1000000 | 0.67 | -0.173925197 |
| 2000000 | 1.66 | 0.220108088 |
| 4000000 | 3.94 | 0.595496222 |
| 8000000 | 9.18 | 0.962842681 |
| 10000000 | 12.38 | 1.092720645 |

original chart with log scale axis

modified chart with logarithmic running time

after calculation we see that the running time are approximately

3.(c) and 4.(c) number of nearest neighbor calculations per second

|  |  |  |
| --- | --- | --- |
|  | 2D Tree | RB Tree |
| Data 1K | 51020.4 | 467.3 |
| Data 1M | 8012.8 | 20.57 |

Note that there is no useful property when doing nearest neighbor search by red-black tree so the way it search are virtually searching all nodes in the tree.