

DM819 - Computational Geometry

Fall 2015

Project 2

Mikkel Levisen and Jesper Lund

December 2, 2015

Contents

| | | |
|----------|---------------------------|----------|
| 1 | Introduction | 1 |
| 2 | Range Queries | 1 |
| 3 | KD-Tree | 1 |
| 3.1 | Complexity | 1 |
| 4 | Range-Tree | 1 |
| 4.1 | Complexity | 1 |
| 5 | Test | 1 |
| 5.1 | Test Generation | 1 |
| 6 | Manual | 4 |
| 6.1 | File Structure | 4 |
| 7 | Conclusion | 5 |

1 Introduction

This report details the implementation of KD-Tree and Range-Tree for n -dimensional input. Each tree is constructed from a list of unsorted points and is capable of performing orthogonal range queries.

2 Range Queries

A point $p \in P$ exists in the range R iff.

$$\forall p_i \in p : \{R_{i,1} \leq p_i \leq R_{i,2}\} | \forall i \in \{1, \dots, n\}, n = \text{dimensions}$$

A range query consists of reporting all points, p , which lie within a given range, R . An n -dimensional point, P_n , consist of real numbers $\{p_1, p_2, \dots, p_n\}$, and an n -dimensional range R_n consist of $[x : x'] \times [y : y'] \times \dots \times [z : z']$. A range query consist of reporting all points P_n in a range R_n .

3 KD-Tree

3.1 Complexity

4 Range-Tree

A Range-tree is a multi-level data structure for time efficient range queries. An n -dimensional Range-tree consist of n levels where each level, l , is a 1-dimensional range query on the l 'th determining canonical subset $P(v)$ associated structure n -level tree

4.1 Complexity

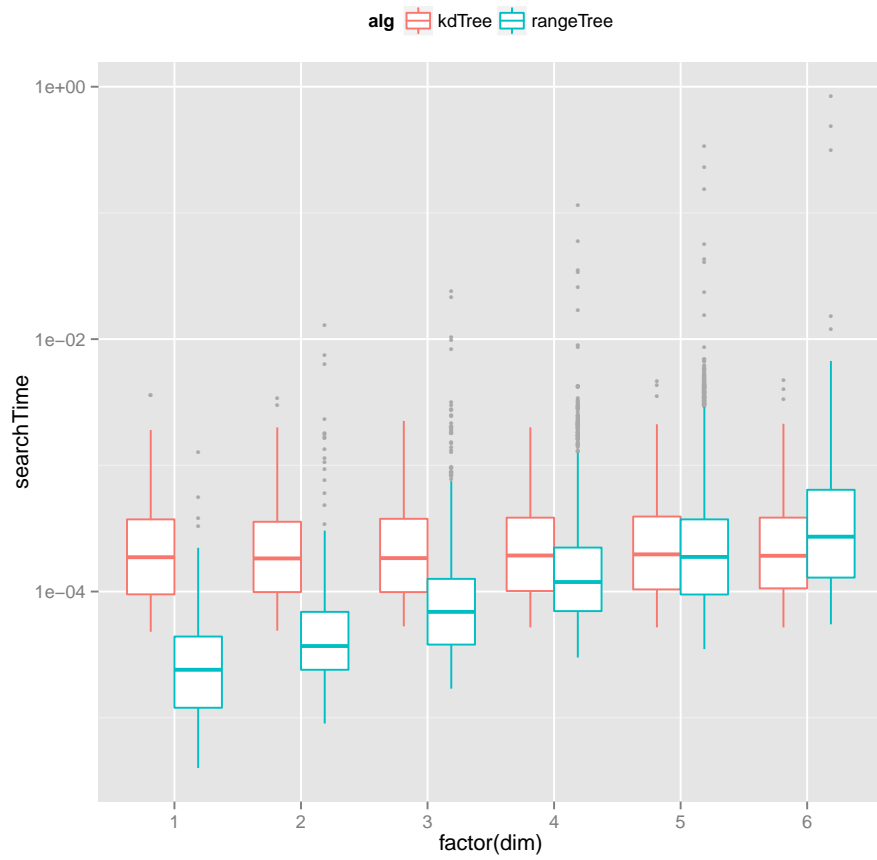
5 Test

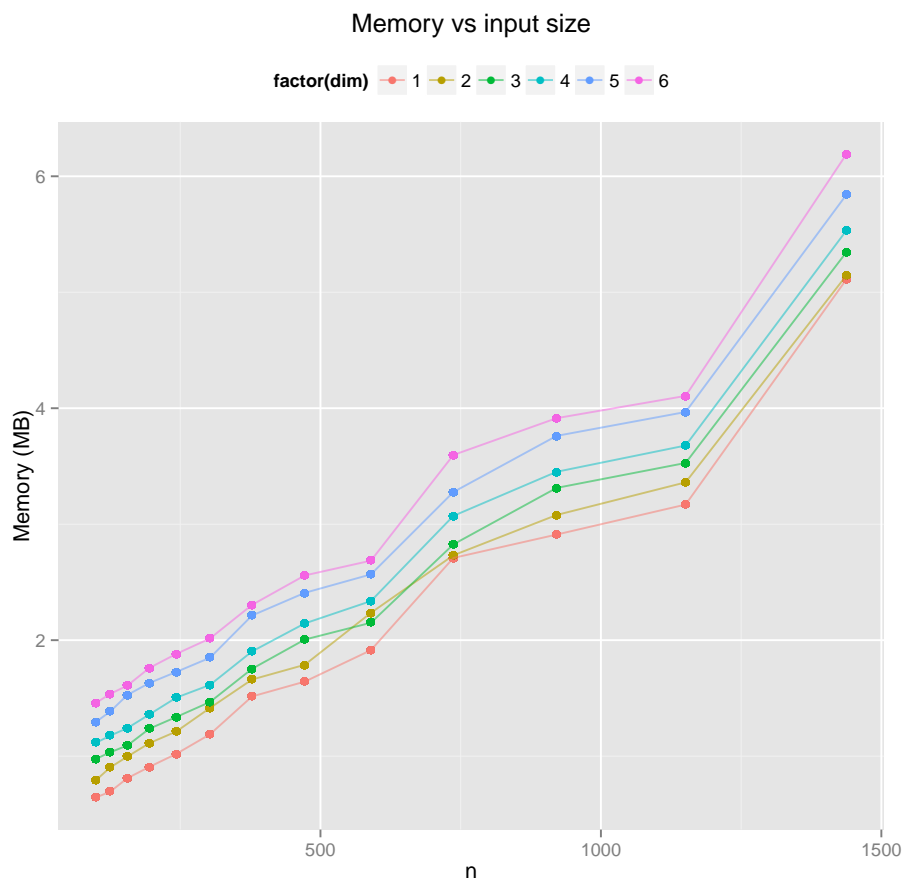
5.1 Test Generation

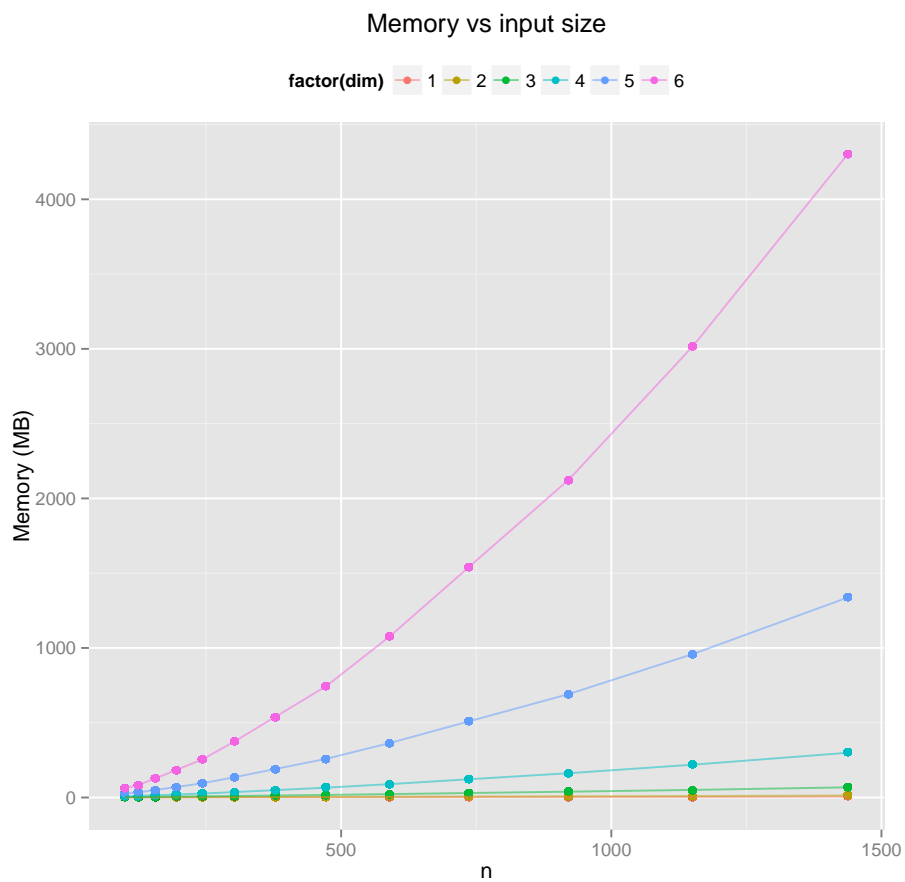
Supplied with the code is a test generating script which can create tests of arbitrary dimension and size. The tests are sorted in directories by dimension.

In a tests consisting of n points in d dimensions an d -dimensional volume is constructed with a side length $s = \sqrt[d]{n}$. For each such volume is 1000 ranges generated with a side length of $rs = \sqrt[d]{0.1 \cdot s^d}$ i.e. each range will contain approximately 10% of the n points.

The generated tests also contains the correct output of each range query.







6 Manual

6.1 File Formats

```

<real> \t <real>
<real> \t <real>
<real> \t <real>
<real>:<real> \t <real>:<real> , <integer> \t <integer>
<real>:<real> \t <real>:<real> , <integer> \t <integer>
<real>:<real> \t <real>:<real>

```

6.2 Inscrutions

6.3 File Structure

```
ROOT
|-- report/
'-- src
    |-- kdtree
    |   |-- inspect.lua
    |   |-- kdtree.lua
    |   '-- test.lua
    |-- R
    |   '-- makePlots.R
    |-- rangeTree
    |   |-- inspect.lua
    |   |-- middleclass
    |   |   '-- middleclass.lua
    |   |-- RangeTree.lua
    |   '-- test.lua
    |-- results/
    |-- runTests.py
    '-- tests
        |-- createCustomTest.lua
        |-- dimension_1/
        |-- dimension_2/
        |-- dimension_3/
        |-- dimension_4/
        |-- dimension_5/
        |-- dimension_6/
        |-- genTestSuite.lua
        '-- inspect.lua
```

7 Conclusion