

# Complex extension: Implementing a B-Tree index

INFO-H417: Lab Session 4

2023-2024

The goal of this session is to implement a B-Tree index for the Complex extension from Session 3. The B-Tree will order and compare the complex numbers by their absolute value (or modulus)  $r = |a + bi| = \sqrt{a^2 + b^2}$ . This can be done in four steps.

1. Implement the 5 comparison functions:<sup>1</sup> equals, less than, less or equals, greater than, greater or equals.
2. Implement the corresponding comparison operators:<sup>2</sup> =, <, <=, >, >=
3. Implement the order support function<sup>3</sup> *complex\_cmp(c,d)*, which returns  $-1$  if  $|c| < |d|$ ,  $0$  if  $|c| = |d|$  or  $+1$  if  $|c| > |d|$ .
4. Implement the B-Tree operator class, using the functions and operators defined previously.<sup>4</sup>

You can find useful information in the footnotes defined at each step. To test the implementation, create a table filled with random complex numbers and build a b-tree index on the column containing the complex values. The index should then be used in subsequent **SELECT** queries that use one of the comparison operators (=, <, <=, >, >=). For example, see the following code.

```
CREATE TABLE t(c complex);
INSERT INTO t(c) SELECT complex(random(), random()) FROM generate_series(1, 1e6);
EXPLAIN ANALYZE SELECT 4 - count(c) * 4 / 1e6 FROM t WHERE c > '(1, 0)';
CREATE INDEX t_idx ON t(c);
EXPLAIN ANALYZE SELECT 4 - count(c) * 4 / 1e6 FROM t WHERE c > '(1, 0)';
```

---

<sup>1</sup><https://www.postgresql.org/docs/current/sql-createfunction.html>

<sup>2</sup><https://www.postgresql.org/docs/current/sql-createoperator.html>

<sup>3</sup><https://www.postgresql.org/docs/current/btree-support-funcs.html>

<sup>4</sup><https://www.postgresql.org/docs/current/xindex.html>