Programming Assignment: B+-Tree

- Implement a B+-Tree
- B+-Tree should work on fixed-size key types and a comparator
- Use the buffer manager to access pages
- B+-Tree must be thread-safe (see slides “Concurrent Access”)
- Determine suitable value for $k$ (different for leaf and inner nodes!)
- Supported operations: point lookup, range lookup
B+-Tree Insertion (no concurrency)

Insert 42 and 70:

| 30 | 55 |  |
B+-Tree Insertion (no concurrency)

Insert 42 and 70:

| 30 | 42 | 55 | 70 |
B+-Tree Insertion (no concurrency)

Insert 15:

| 30 | 42 | 55 | 70 |

Moritz Sichert (TUM) | Database Systems on Modern CPU Architectures
B+-Tree Insertion (no concurrency)

Insert 15:
B+-Tree Insertion (no concurrency)

Insert 15:

```
15 30 42 55 70
```

Diagram shows a B+-tree node with keys 15, 30, 42, 55, 70.
B+-Tree Insertion (no concurrency)

Insert 71-79:

```
  42
/    \
15 30 42  55 70
```

Moritz Sichert (TUM) | Database Systems on Modern CPU Architectures
B+-Tree Insertion (no concurrency)

Insert 71-79:
B+-Tree Insertion (no concurrency)

Insert 61, 62:
B+-Tree Insertion (no concurrency)

Insert 61, 62:
B+-Tree Insertion (no concurrency)

Insert 61, 62:

```
  15 30 42
  |     |
  |     |
  55 61 62 70 71
  |     |
  72 73 74
  |     |
  74 77
  |     |
  74 77
  |     |
  78 79
```
B+-Tree Insertion (no concurrency)
Concurrent B+-Tree Insertion

Insert 16:
Concurrent B+-Tree Insertion

Insert 16:
Concurrent B+-Tree Insertion

Insert 16:
Concurrent B+-Tree Insertion

Insert 16:
Concurrent B+-Tree Insertion

Insert 16:
Concurrent B+-Tree Insertion

Insert 16:
Concurrent B+-Tree Insertion

Insert 16:
Concurrent B+-Tree Insertion

Insert 16:
Concurrent B+-Tree Insertion

Insert 16:

10 20 30 40 70 80

14 16 19 25 50