



# R-Trees

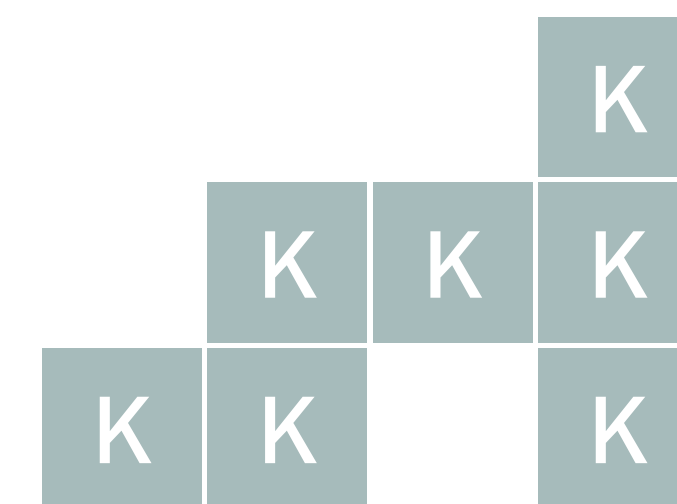
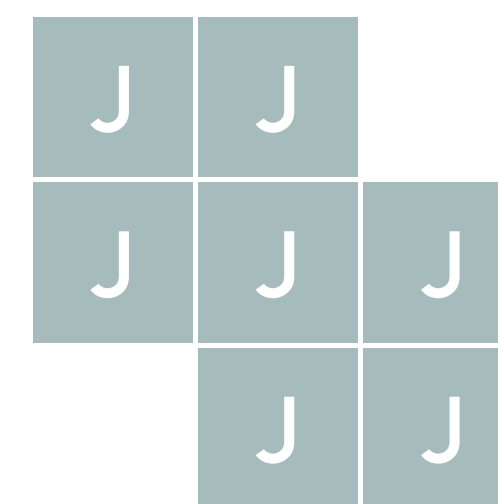
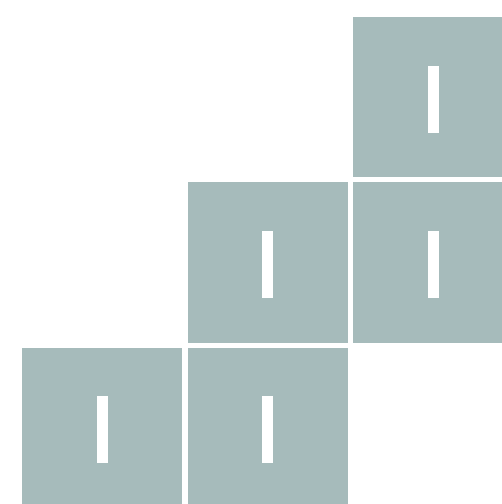
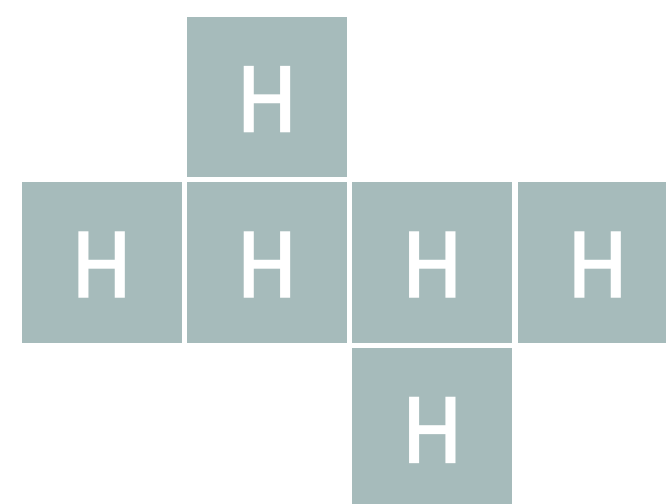
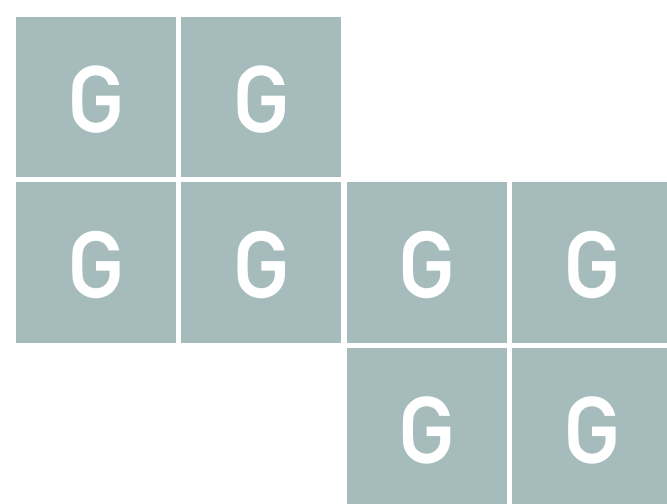
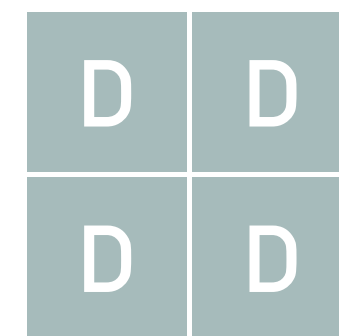
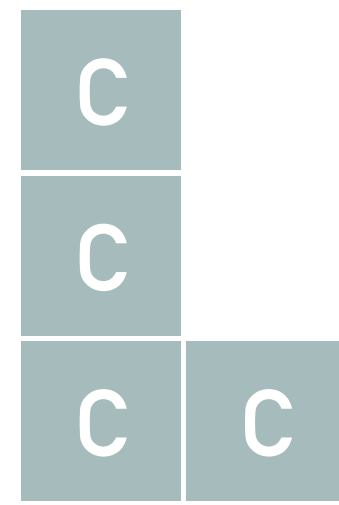
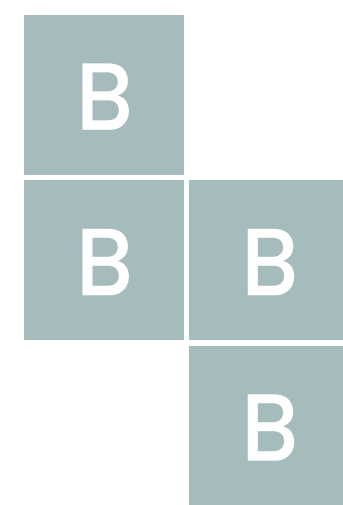
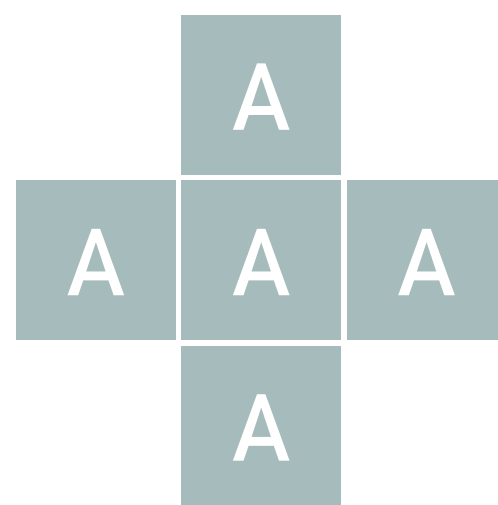
---

*NDBI007: Assignment 5*

# Task 1

---

- ❖ Insert the following 11 elements arbitrarily into a node with dimensions  $a = \langle 8, 14 \rangle$ ,  $b = \langle 8, 14 \rangle$ ,  $a \neq b$
- ❖ Elements can be rotated and overlapped freely
- ❖ You will use this node in the following tasks



❖ **Points: 0**

# Task 2: Guttman's Split

---

- ❖ Split the overflowed node with *Guttman's method*
  - ❖ The maximum number of items in a node is  $M = 10$
  - ❖ The minimum number of items in a node is  $m = \langle 3,5 \rangle$
  - ❖ In the PickSeeds method, calculate the overall area, area of the elements, and dead space *only for the 4 selected most distant pairs* of elements
    - ❖ I.e., you do not need to consider all possible pairs of elements
- ❖ *Illustrate* all the steps of splitting a node and the result
- ❖ Or, you may *implement* the algorithm *and log* all events (i.e., submit the source code and, e.g., makefile)
  - ❖ The permitted languages are Java, Python, C, C++, and Swift
- ❖ **Points: 1**

# Task 3: Greene's Split

---

- ❖ Split the overflowed node with *Greene's method*
  - ❖ The maximum number of items in a node is  $M = 10$
  - ❖ The minimum number of items in a node is  $m = \langle 3, 5 \rangle$
  - ❖ In the PickSeeds method, calculate the overall area, area of the elements, and dead space *only for the 4 selected most distant pairs* of elements
    - ❖ I.e., you do not need to consider all possible pairs of elements
- ❖ *Illustrate* all the steps of splitting a node and the result
- ❖ Or, you may *implement* the algorithm *and log* all events (i.e., submit the source code and, e.g., makefile)
  - ❖ The permitted languages are Java, Python, C, C++, and Swift
- ❖ **Points: 1**

# Task 4: Splitting In R\* Tree

---

- ❖ Split the overflown node with *R\* Tree split method*
  - ❖ The maximum number of items in a node is  $M = 10$
  - ❖ The minimum number of items in a node is  $m = \langle 3,5 \rangle$
- ❖ *Illustrate* all the steps of splitting a node and illustrate the result
- ❖ Or, you may *implement* the algorithm *and log* all events (i.e., submit the source code and, e.g., makefile)
  - ❖ The permitted languages are Java, Python, C, C++, and Swift
- ❖ **Points: 1**

# Bonus Task 5

---

- ❖ *Compare* the individual *methods* of splitting of an overflowed node
  - ❖ I.e., Guttman's, Greene's, and R\* Tree split
  - ❖ What are the *advantages* and *disadvantages*?
  - ❖ What *effect* do they have on the *height* or *occupancy* of the R(\*) tree?
  
- ❖ **Points: 1**