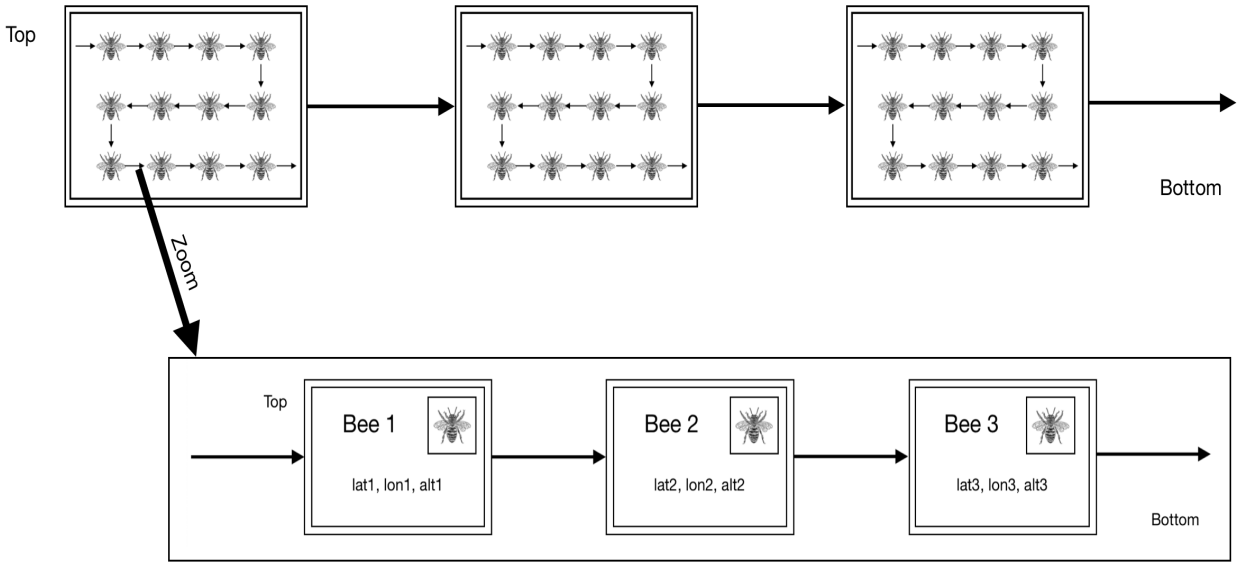
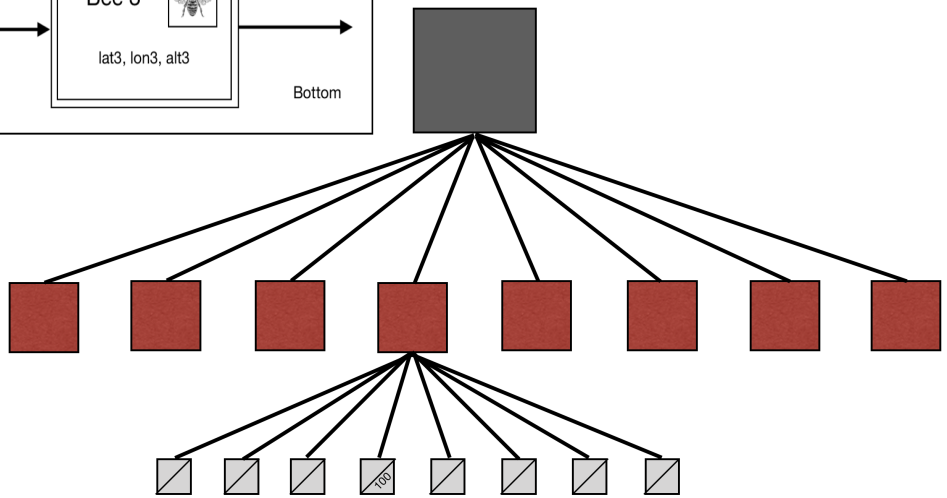
# Collision prevention with 3D coordinates

By: Daniel Otero Gómez
Rafael Mateus Carrión

# Data Structure



# Complexity

| Method | Complexity |
|---|---|
| leer | O(n) |
| splitString | O(1) |
| getMaxMin | O(1) |
| choque | O(n) |
| octree | O(n) |
| hashing | O(1) |
| nuevoOctree | O(1) |

# Why is our data structure a good solution?

- Our data Ssructure makes an excellent use of structures that Java provide us.
- It saves up steps in order to get to the solution.
- The code, as the object created by us to represent the bees, is simple enough to be understood by everyone and to get as near to the optimal solution as possible.

# Time and Memory Usage Tables

| Data Set | Time (ms) |
|---|---|
| 4 | 0 |
| 10 | 0 |
| 15 | 0 |
| 100 | 3 |
| 150 | 2 |
| 1000 | 9 |
| 1500 | 11 |
| 10000 | 9 |
| 15000 | 12 |
| 100000 | 174 |
| 150000 | 293 |
| 1000000 | 2095 |
| 1500000 | 453 |

| Data Set | MemoryUsed (MB) |
|---|---|
| 4 | 0 |
| 10 | 0 |
| 15 | 0 |
| 100 | 1 |
| 150 | 3 |
| 1000 | 10 |
| 1500 | 59 |
| 10000 | 24 |
| 15000 | 39 |
| 100000 | 107 |
| 150000 | 151 |
| 1000000 | 227 |
| 1500000 | 126 |

# Image of the data structure running