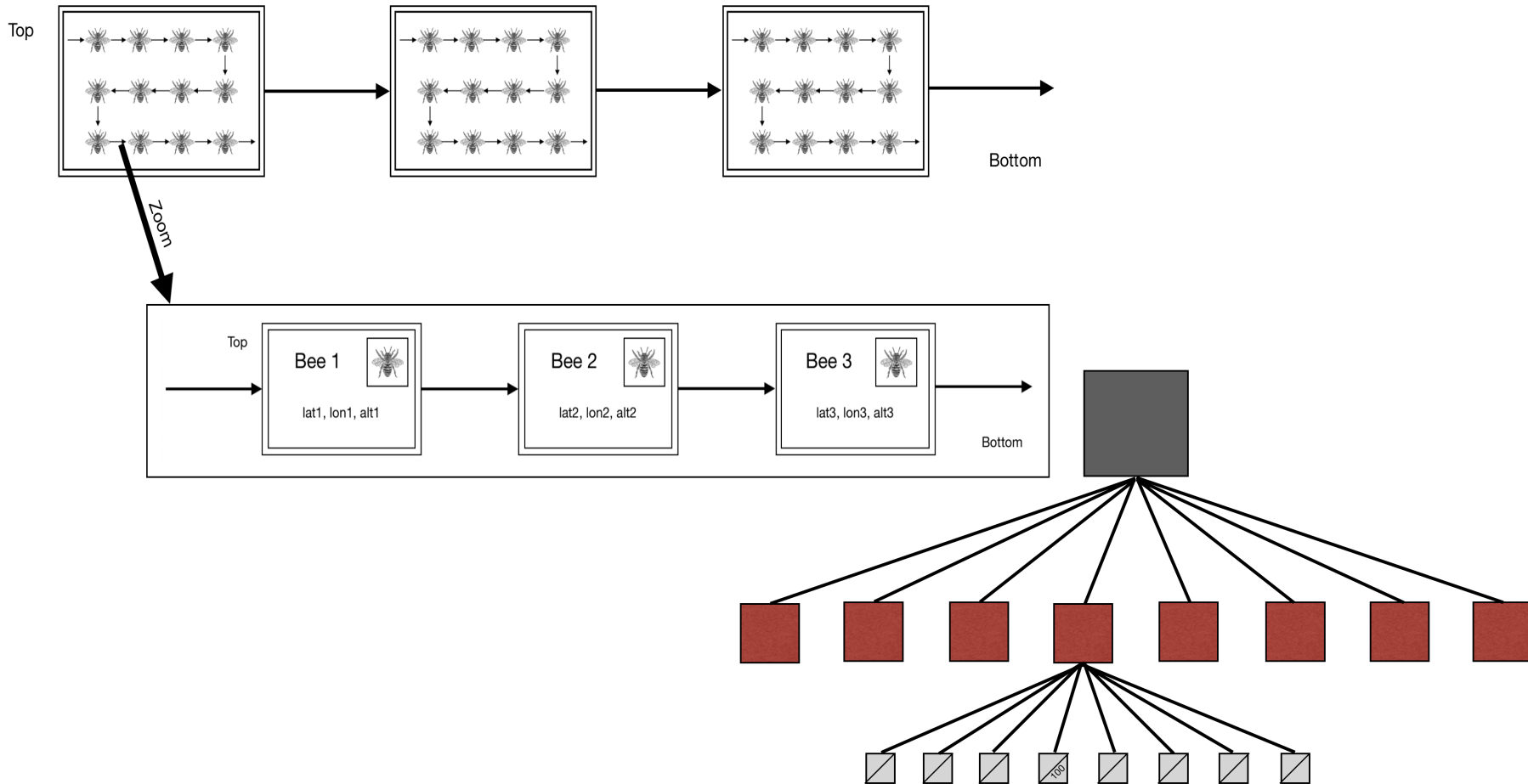


Collision prevention with 3D coordinates

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 Sstructure 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

```
-73.6807366644, 2.99504342745, 1723.131
-72.09517556722142404258628, 1690.88
-76.7330962288, 2.62436555581, 1722.65
-75.6772888293, 2.18334762599167319.95
-74.6313928698, 2.90509520666, 1629.98
-73.78138069449, 2.75409483381, 1748.65
-71.8650720999, 2.7169326983, 1773.834
-75.4576466963, 2.40061717747, 1570.66
-71.8845063568, 2.84339948269, 1732.47
-72.0567236623, 2.08275363097, 1647.75
-75.7453203644, 2.85430778425, 1633.62
-75.2962220022, 2.93588993468, 1784.16
-75.5326485982, 2.36152572359, 1762.21
-72.7162988276, 2.7377401937, 1678.658
-75.948244174932096782217675, 1693.96
-75.8206372397, 2.82986000374, 1771.86
-74.9875998023, 3.31402998761, 1572.29
-72.3326760879, 2.16723734467, 1788.26
-73.0206995018, 2.68815864619, 1559.03
-73.9621438027, 2.08540415879, 1741.36
-72.4260549498, 2.03645369928, 1609.63
-75.9624925782, 2.62593549319, 1636.33
-75.3060429747
```

Type input and press Enter to send to program