

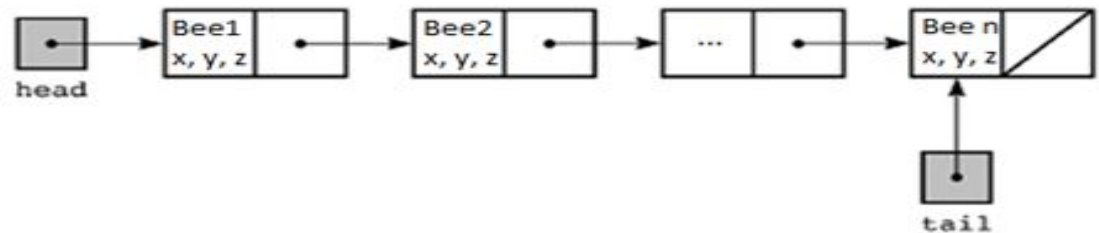
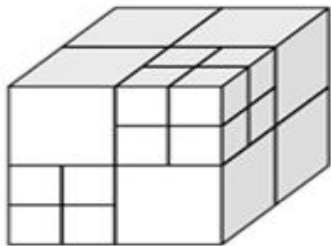
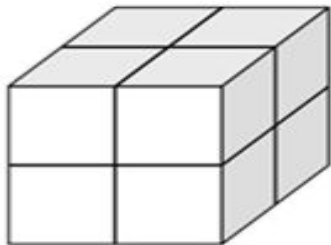
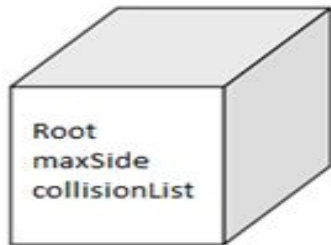
POLLINATORS DRONES: A COLLISION DETECTION ALGORITHM

Anthony García

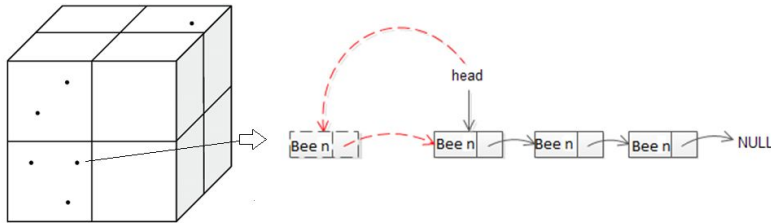
Daniel Hincapié

Medellín, May 14th, 2019

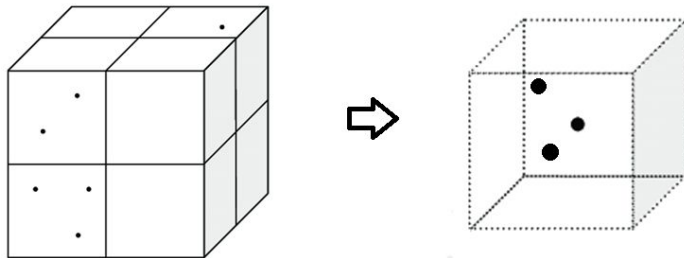
Design Data Structure: Octree



Data Structure Operations



Grafic 2: Insert operation of an octree



Grafic 3: Detect Collisions operation of an octree

Operation	Complexity
Read File	$O(n)$
Insert	$O(\log(s))$
Detect collisions	$O(n*s)$
Write File	$O(n)$

Table 1: Complexity of the operations

Design Criteria of the Data Structure

- ▣ Octrees are really efficient, they allow us to implement the operations in a low complexity. It give us optimization, speed and efficiency.
- ▣ When the octree is implemented with linkedList, the insertion complexity is really good. $O(\log(s))$ (Where s is the space).
- ▣ Is an easy to use structure.
- ▣ It is ideal to be used in the detection of collisions.

Time and Memory Consumption

Operation	100 bees	1000 bees	100000 bees	1000000 bees
Read File	6 ms	6 ms	72 ms	571 ms
Insert	3 ms	2 ms	41 ms	368 ms
Detect Collisions	0 ms	0 ms	1 ms	1 ms
Write	5 ms	3 ms	117 ms	954 ms

Time for bees in Bello

Memory Consumption	100 bees	1000 bees	100000 bees	1000000 bees
	1,63 MB	3 MB	24,25 MB	166,5 MB
	150 bees	1500 bees	150000 bees	1500000 bees
	2 MB	4,4 MB	188,8 MB	1436 MB

Time for bees in Colombia

Memory consumption