THREE-DIMENSIONAL MATRICES IMPLEMENTED WITH LINKED LISTS TO PREVENT COLLISIONS

Juan Sebastián Pérez Salazar

Yhoan Alejandro Guzmán Garcia

Medellín, (November 6th)

data structure designed

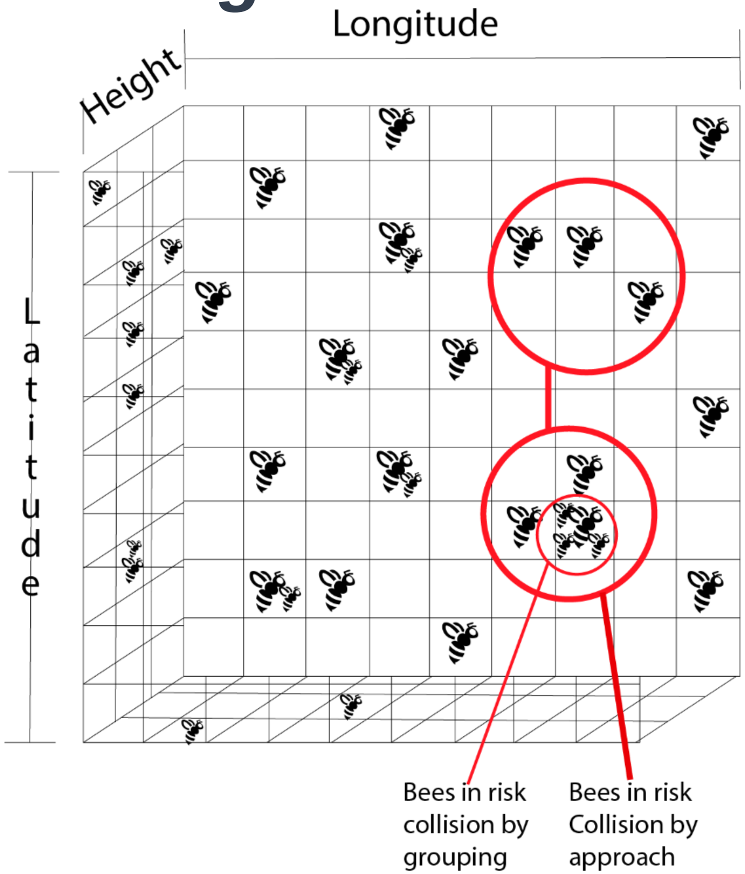
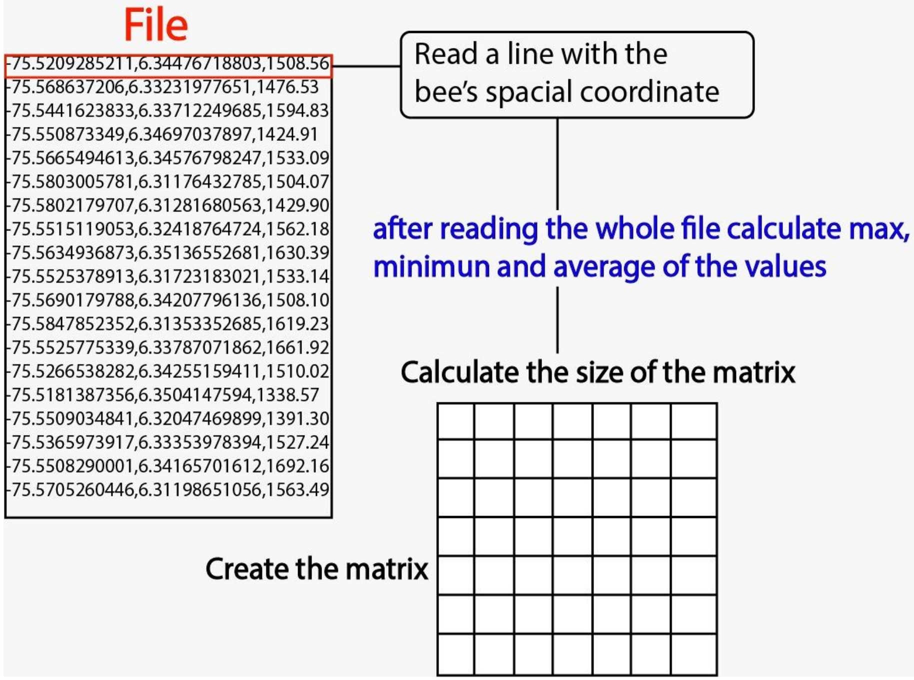

Longitude
Height
Latitude
Bees in risk collision by grouping
Bees in risk Collision by approach

Figure 1: Prevention process of bees in collision risk, from Three-Dimensional Matrices implemented with LinkedLists.

Data Structure Operations



File

```
-75.5209285211,6.34476718803,1508.56
-75.568637206,6.33231977651,1476.53
-75.5441623833,6.33712249685,1594.83
-75.550873349,6.34697037897,1424.91
-75.5665494613,6.34576798247,1533.09
-75.5803005781,6.31176432785,1504.07
-75.5802179707,6.31281680563,1429.90
-75.5515119053,6.32418764724,1562.18
-75.5634936873,6.35136552681,1630.39
-75.5525378913,6.31723183021,1533.14
-75.5690179788,6.34207796136,1508.10
-75.5847852352,6.31353352685,1619.23
-75.5525775339,6.33787071862,1661.92
-75.5266538282,6.34255159411,1510.02
-75.5181387356,6.3504147594,1338.57
-75.5509034841,6.32047469899,1391.30
-75.5365973917,6.33353978394,1527.24
-75.5508290001,6.34165701612,1692.16
-75.5705260446,6.31198651056,1563.49
```

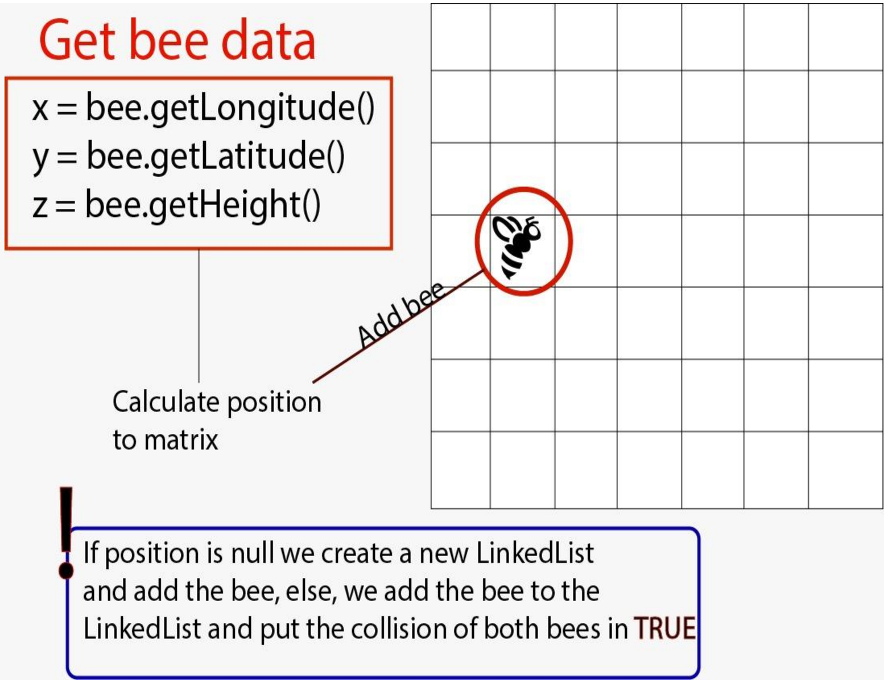
Read a line with the bee's spacial coordinate

after reading the whole file calculate max, minimun and average of the values

Calculate the size of the matrix

Create the matrix

Figure 2: Process of analysis and reading of data



Get bee data

```
x = bee.getLongitude()
y = bee.getLatitude()
z = bee.getHeight()
```

Calculate position to matrix

If position is null we create a new LinkedList and add the bee, else, we add the bee to the LinkedList and put the collision of both bees in TRUE

Figure 3: Process of adding bees to the matrix

Data Structure Operations

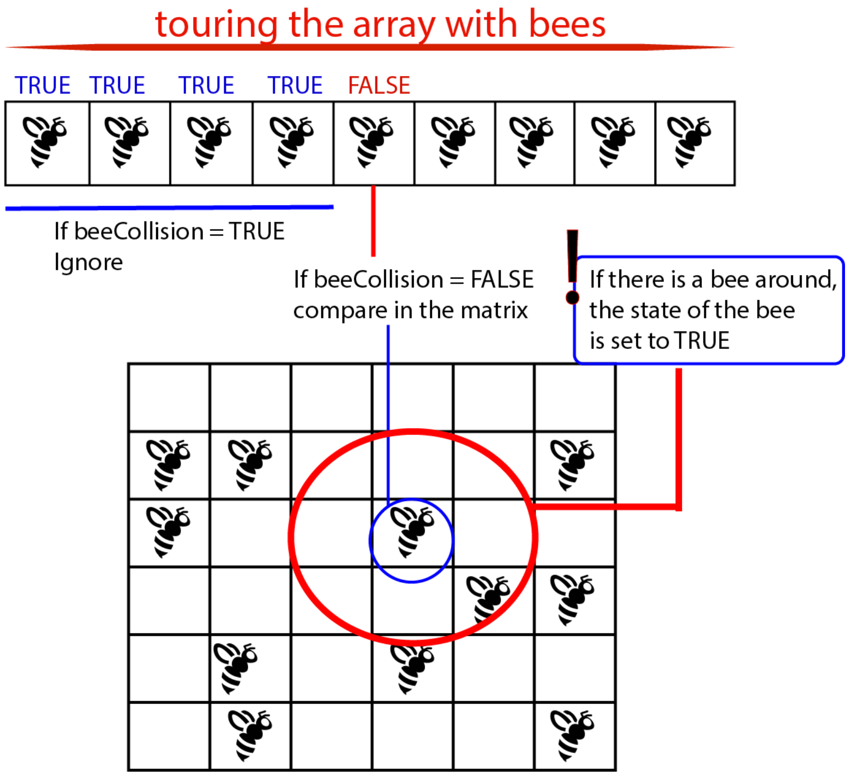


Figure 2: Process of detecting possible collisions between bees

Method	Complexity
Analysis and reading of data	O(n)
Add bees to matrix	O(n)
Detect possible collisions	O(n)
SaveFile	O(n)

Table 1: Complexity of operations of the data structure

Design Criteria of the Data Structure

- It is a highly effective data structure for what is sought.
- It has a multiple implementation.
- This structure compensates the memory expense with a high speed

When you have millions of bodies at collision risk, you should keep in mind that the first thing is to give quick answers.

Time and Memory Consumption

Data set (Number of bees)	Best time (ms)	Worst time (ms)	Average time (ms)
4	0	0	0
10	0	0	0
100	1	4	2
1000	7	9	8
10000	13	13	13
100000	35	37	36
1000000	195	198	196

Table 2: Execution time (in milliseconds) of the data structure for each data set .

Data set (Number of bees)	Best memory (mb)	Worst memory (mb)	Average memory (mb)
4	0	0	0
10	0	0	0
100	0	0	0
1000	0	0	0
10000	2	2	2
100000	4	4	4
1000000	25	28	26

Table 3: Memory used (in megabytes) by the data structure for each data set.

Implementation

(Video)