Comparing data structures

Data structures perform differently for the insertion and selection of data.

Experiment

A simple experiment has been set up to gather experimental data:

https://github.com/ros101/data-structures-comparison

The experiment consist in inserting unordered key-value pairs into various structures. One random key is then used to measure the performances to return a value. The experiment also checks the correctness of the returned value. No failures have been detected.

Result

The following is an example of the result obtained from the experiment. Multiple executions are consistent with this report.

```
type           time to insert time to find   fail
ArrayStructure 43,993,376     9,033,290      False
LinkedList     145,084,444    9,202,950      False
BTree          796,571,810    669,562        False
HashMap        233,445,188    258,856        False
```

Time is expressed in CPU cycles.

Review of the results

As expected Arrays are by far the most efficient structure to store data, but they are also very inefficient when it is time to find a specific value.

Overall, linked lists perform worse than Arrays in insertion and comparably in selection.

Btree is the most complex structure and they have an expensive insertion process, but they are very efficient in retrieving data.

Hash maps effectively optimize linked lists resulting in slightly less efficient insertion but the fastest structure to retrieve data from.

Key takeaways

- arrays should be preferred every time there is no need to search for a specific entry.
- linked lists should be used as a last resort to replace arrays when it is not known the number of elements that they should contain.
- B-trees are recommended when it is important to give priority to searching for an element.
- Hash maps are a balanced choice when it is possible to define an index for the data.

Other structures

Queues and stack do not fit the use case of this experiment, but, given their nature, they would perform like a linked list.