**Introduction**

This report evaluates the performance of four common data structures: Linear List, Balanced Binary Search Tree (BBST), Trie, and Hash Map. By comparing their build times and text check times, we can determine which data structure is most suitable for different applications. The results are visualized in bar graphs to provide a clear comparison of their performance.

**Description of Algorithms**

***Linear List***

This is a simple list where each item is stored one after the other. Searching takes longer as it checks each item one by one. Building (adding items) is quick.

***Balanced Binary Search Tree (BBST)***

A special tree structure that keeps items balanced for quicker searching. Searching and adding items are fairly quick. Building the tree takes some time to keep it balanced.

*Trie*

A tree-like structure where each node represents part of a word. Great for searching words quickly. Building it takes a lot of time because each part of each word needs to be added.

*Hash Map*

Uses a hash function to map keys to values. Very fast for searching and adding items. Building it is also quite fast.

**Findings**

We looked at two main things: how long it takes to build (set up) the dictionary and how long it takes to check (search) text.

***Dictionary Build Time****:* **Linear List** takes almost no time to build. **BBST** takes a moderate amount of time to build. T**rie** takes the longest time to build and **Hash Map** builds quite fast.

***Text Check Time****:*The Linear List takes a very long time to check text, Hash Map is the fastest for checking text, BBST and Trie are both fast, with the BBST being slightly quicker.

Linear List - Build time: 0.0000s, Check time: 31.4518s

BBST - Build time: 0.0592s, Check time: 0.0099s

Trie - Build time: 0.4069s, Check time: 0.0307s

Hash Map - Build time: 0.0186s, Check time: 0.0021s

Изображение выглядит как График, линия, диаграмма, Прямоугольник

Автоматически созданное описание