

TRABALHO DE ESTRUTURA DE DADOS 2 COMPARATIVO DE DESEMPENHO DE BUSCA ENTRE ESTRUTURAS

NOME: GABRIEL ALESSI POSONSKI || RA: 2259583

NOME: GABRIEL VITOR SIQUEIRA || RA: 2201631

Este trabalho tem como objetivo avaliar o desempenho entre as estruturas de dados AVL e Hash, sendo o último nos formatos dinâmico e estático

Para os arquivos disponibilizados via Moodle, obteve-se os seguintes resultados:

AVL

```
Tempo de execucao (in1.txt): 0.012 segundos
Total de comparacoes: 4560
Tempo de execucao (in2.txt): 0.012 segundos
Total de comparacoes: 4606
Tempo de execucao (in3.txt): 0.012 segundos
Total de comparacoes: 4950
```

Hash Dinâmico

```
=====Arquivo 1=====
Tempo de busca em segundos: 0.000000
Comparacoes: 500
=====Arquivo 2=====
Tempo de busca em segundos: 0.015000
Comparacoes: 502
=====Arquivo 3=====
Tempo de busca em segundos: 0.000000
Comparacoes: 37325
=====
Tempo total de execucao em segundos: 0.015000
```

Hash Fixo

```
=====Arquivo 1=====
Tempo de busca em segundos: 0.000000
Comparacoes: 2821
=====Arquivo 2=====
Tempo de busca em segundos: 0.000000
Comparacoes: 2811
=====Arquivo 3=====
Tempo de busca em segundos: 0.000000
Comparacoes: 18125
=====
Tempo total de execucao em segundos: 0.000000
```

Respondendo às questões solicitadas:

- 1) O pior caso do hash dinâmico foi o terceiro arquivo.
O melhor caso foi para os arquivos in1 e in2, tendo poucas diferenças entre si.
- 2) O pior caso foi o arquivo in3
O melhor desempenho foi localizado no arquivo in1.
- 3) Bônus: o pior caso para o hash fixo também foi o arquivo 3,
Os melhores casos foram para os 2 primeiros arquivos, também possuindo pouca diferença de desempenho.
Quanto ao tempo de execução, ambos tiveram valores muito inconstantes, variando abaixo de 0.016 segundos e em algumas execuções o contador não conseguiu encontrar algoritmos significativos, então não é possível tirar conclusões de desempenho, baseando-se no tempo, mas notoriamente arquivos que demandarem mais comparações terão um tempo de execução mais longo.
- 4) De maneira isolada, pode-se dizer que os hash's possuem um desempenho melhor nos 2 primeiros arquivos atingindo o caso ideal de $O(1)$, porém isso não ocorre no arquivo 3, já que houveram várias colisões com as chaves inseridas, ocasionando em um custo de busca maior para a maioria das buscas. No caso da AVL, independente do arranjo de números, obteve-se uma quantidade de comparações semelhantes em todos os arquivos, mostrando que a capacidade de auto-balanceamento otimiza os casos de busca, onde teremos o desempenho $O(\log n)$. Portanto, ao analisar de maneira geral, pode-se considerar a AVL como uma estrutura mais otimizada para buscas.