

Universidade de São Paulo  
Instituto de Matemática e Estatística  
Bacharelado em Ciência da Computação

Proposta de Trabalho de Formatura Supervisionado  
MAC0499

# Árvores Binárias de Busca e a Conjectura da Otimalidade Dinâmica

Bruno Armond Braga

Orientadora: Profa. Dra. Cristina Gomes Fernandes

## 1 Introdução

Árvores Binárias de Busca (ABBs) são estruturas de dados que armazenam um conjunto de chaves de um universo estático, que possui uma ordem total, e dão suporte a buscas neste conjunto. Essas estruturas são fundamentais para o universo da computação.

Padrões de acesso do mundo real muitas vezes possuem estruturas repetitivas, como por exemplo bancos de dados que recebem solicitações frequentes para um pequeno número de elementos de alto tráfego. Nesse estudo estaremos interessados em explorar as delimitações de custo dos algoritmos de busca em ABBs.

Apesar de existirem ABBs muito bem documentadas com tempo por busca logarítmico no número de chaves armazenadas, essas estruturas normalmente não conseguem alcançar uma eficiência superior a isso independentemente da entrada.

ABBs *online* são ABBs que apenas possuem informações sobre os acessos passados. Ou seja, ao acessar  $x_i$ , o algoritmo de busca não tem conhecimento das chaves  $x_{i+1}, \dots, x_m$ , nem mesmo do valor de  $m$ .

Seja  $\text{OPT}(X)$  o menor custo possível de algum algoritmo de busca em ABB que acessa todas as chaves da entrada  $X$ . Uma ABB *online* é *dinamicamente ótima* se, para todas as sequências  $X$ , seu algoritmo de busca tem custo  $O(\text{OPT}(X))$ . Considera-se que não estão ocorrendo inserções ou remoções no conjunto durante estes acessos.

Todo esse estudo é feito para tentar responder à pergunta: “*Existe uma ABB online dinamicamente ótima?*”.

Uma das tentativas de responder tal pergunta foi a árvore splay de Sleator e Tarjan [7]. Árvores splay são ABBs que seguem a heurística “move to front”. Assim, após cada acesso, a árvore se reestrutura de uma maneira particular, trazendo o nó da chave que foi acessada para a raiz da árvore.

Há uma conjectura não resolvida no artigo [7] que diz que as árvores splay são dinamicamente ótimas. Essa conjectura ficou conhecida como a *Conjectura da Otimalidade Dinâmica*.

## 2 Objetivos

Os problemas da atualidade são cada vez mais desafiadores, envolvendo quantidades enormes de dados, e requerendo técnicas algorítmicas mais eficientes e elaboradas para suas resoluções. As aplicações também se mostram cada vez mais sofisticadas, exigindo o desenvolvimento de teorias e conceitos que se adéquem às novas situações e aos novos desafios. Por outro lado, as novas técnicas por vezes possibilitam a resolução de antigos problemas de maneiras novas, mais interessantes e elegantes. Por estes motivos, o estudo e desenvolvimento de temas ligados a estruturas de dados é uma área que permanece ativa ao longo dos anos, com inovações que têm reflexos em diversas subáreas da computação e também em outras áreas.

O objetivo deste trabalho é aprofundar e consolidar os conhecimentos em estruturas de dados tradicionais, bem como o estudo de tópicos sofisticados envolvendo uma estrutura de dados clássica na computação: as árvores binárias de busca.

O tema central do trabalho é a Conjectura da Otimalidade Dinâmica, que diz respeito às chamadas árvores splay. Adicionalmente, um material sobre o tema será desenvolvido explicando de maneira didática as análises dos algoritmos e estruturas de dados estudadas.

## 3 Metodologia

Serão utilizados como base de estudos os artigos e livros citados ao longo de todo o projeto. O aluno irá interagir ativamente com a supervisora. Um dos principais aspectos dessa interação envolverá a redação de partes do estudo e material de pesquisa, além da avaliação das implementações desenvolvidas. Essa prática serve para diversos propósitos, incluindo avaliar o nível de compreensão do aluno e permitir que a supervisora forneça feedback e orientação pontuais conforme necessário. Esse processo iterativo de feedback garantirá que a abordagem de pesquisa do candidato permaneça no rumo certo e esteja alinhada com os objetivos do projeto, e também garantirá que as implementações legíveis, bem documentadas e de acordo com suas especificações.

## 4 Cronograma

O cronograma estipulado é o seguinte:

| Ativ/Mês | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|----------|---|---|---|---|---|---|---|----|----|
| 1        | ✓ | ✓ |   |   |   |   |   |    |    |
| 2        |   | ✓ | ✓ | ✓ |   |   |   |    |    |
| 3        |   |   | ✓ | ✓ | ✓ |   |   |    |    |
| 4        |   |   |   |   | ✓ | ✓ | ✓ | ✓  |    |
| 5        |   |   |   |   |   |   | ✓ | ✓  | ✓  |
| 6        | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓  | ✓  |
| 7        |   |   |   |   |   |   |   | ✓  | ✓  |

### Legenda:

1. Estudo do modelo de computação, da conjectura da otimalidade dinâmica e das árvores splay.
2. Estudo da interpretação geométrica dos algoritmos de busca em ABBs.
3. Estudo do algoritmo guloso proposto em [3].
4. Estudo das delimitações de Wilber.
5. Estudo das árvores tango.
6. Escrita da monografia
7. Preparação do pôster e finalizações do site

## 5 Referências

### Referências

- [1] Tyrel Clayton. BST comparison. Disponível em <https://sites.google.com/view/comparison-dynamic-bst/>. Acesso em 15/04/2024.
- [2] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 2009.
- [3] Erik D. Demaine, Dion Harmon, John Iacono, Daniel M. Kane, and Mihai Patrascu. The geometry of binary search trees. In *ACM-SIAM Symposium on Discrete Algorithms*, 2009.
- [4] Erik D. Demaine, Dion Harmon, John Iacono, and Mihai Pătraşcu. Dynamic optimality — Almost. *SIAM Journal on Computing*, 37(1):240–251, 2007.
- [5] Caleb C. Levy and Robert E. Tarjan. A foundation for proving splay is dynamically optimal, 2022. DOI: 10.48550/arXiv.1907.06310.
- [6] Robert Sedgewick and Kevin Wayne. *Algorithms*. Addison-Wesley Professional, 2011.
- [7] Daniel Dominic Sleator and Robert Endre Tarjan. Self-adjusting binary search trees. *J. ACM*, 32(3):652–686, 1985.
- [8] Robert Wilber. Lower bounds for accessing binary search trees with rotations. *SIAM Journal on Computing*, 18(1):56–67, 1989.