

Plano de Trabalho de Conclusão de Curso

Uma biblioteca de Estrutura de Dados para Operações em Intervalos de vetor

UDESC – Centro de Ciências Tecnológicas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação – Integral
Turma 2019/1 – Joinville/SC

Adilson Luis Jonck Junior – jonckjuniorr@gmail.com
Karina Giradi Roggia – karina.roggia@gmail.com (*orientadora*)

15 de março de 2019

Resumo

O objetivo deste trabalho é implementar uma biblioteca de estruturas de dados para operações em intervalos de vetor uma vez que tais operações são frequentes em problemas de competição de programação. A literatura em português para este tema é escassa e portanto o trabalho também suprirá esta demanda. Dessa maneira, o material poderá ser utilizado para capacitar o leitor no domínio dessas operações e também para reutilizar os conceitos e códigos presentes em futuros problemas relacionados a um cenário competitivo, profissional ou acadêmico. Primeiramente será feita uma fundamentação teórica sobre a análise de complexidade, uma fundamentação sobre as estruturas e suas operações possíveis, será elencado também problemas associados às respectivas estruturas no cenário competitivo e em seguida a implementação dos códigos de cada estrutura. Entre as estruturas de dados que são utilizadas para operações em intervalos, destacam-se as seguintes: *Prefix Sum*, *Sparse Table*, *Fenwick Tree*, *Segment Tree*, *Treap*, em ordem de robustez nas capacidades de suas operações.

Palavras-chave: *Estrutura de Dados. Otimização. Operações em Intervalo*

1 Introdução e Justificativa

A maratona de programação é uma competição mundial que ocorre anualmente no Brasil desde 1996. Nela, os participantes são avaliados de acordo com a sua capacidade de trabalhar em equipe, trabalhar sobre pressão, e resolver o maior número de problemas no menor tempo possível, de certa forma simulando uma situação real de profissionais da área (DUARTE, 2017). Entre os temas mais frequentes da competição encontram-se grandes áreas da computação, como: programação dinâmica, algoritmos gulosos, grafos, implementações, geometria e afins, visto que são

temas robustos em seus conteúdos (VERHOEFF, 2009). Entre as diversas áreas da computação, é recorrente o uso de estruturas de dados como a fundamentação de um algoritmo. Neste trabalho em específico serão abordadas as estruturas de dados para operações em intervalos de vetor, pois são amplamente utilizadas dentro das grandes áreas da computação. Entende-se como intervalo de vetor uma sequência contígua de valores na memória, onde esses valores representam algo para o problema.

Apesar do alcance da maratona de programação ter crescido consideravelmente nos últimos anos, a literatura brasileira sobre os tema não seguiu essa proporção, onde a literatura encontrada é majoritariamente estrangeira e escassa (KAWAKAMI, 2017) (TOMMASINI, 2014). Assim, se vê a necessidade de fundamentar esses conceitos e unifica-los em um só lugar para a construção de uma biblioteca relevante que possa capacitar o leitor sobre a fundamentação dos conceitos das estruturas de dados em intervalos e suas aplicabilidades, tanto para competições quanto para futuras situações no cenário profissional ou acadêmico (COUTO, 2016).

Inicialmente será fundamentado o conceito de notação assintótica e a fundamentação das estruturas que frequentemente são utilizadas para operações em intervalos no cenário de competições de programação, em seguida será elencado problemas relacionados a cada uma das estruturas e o código de suas implementações (HALIM et al., 2013). A seguir são listadas as estruturas de dados que serão exploradas no desenvolvimento deste trabalho, por ordem de poder computacional.

1.1 Prefix Sum

A estrutura mais básica, se trata de um vetor de prefixo acumulativo, onde sua função depende do problema, de maneira geral serve como um acumulador. Não suporta modificações depois que é construído, e certas operações como máximo ou mínimo de um intervalo.

1.2 Sparse Table

Semelhante a estrutura anterior, é estático, ou seja, não pode ser modificado depois de construído, porém, é um pouco mais flexível com as suas operações, suporta máximo e mínimo do intervalo.

1.3 Fenwick Tree

É uma estrutura dinâmica, dessa maneira, aceita modificações depois de sua construção. Não suporta operações como máximo e mínimo, assim como o *Prefix Sum*. É limitado nas suas operações, porém, possui constantes de baixa complexidade assintótica, assim como um código conciso.

1.4 Segment Tree

É uma estrutura dinâmica assim como a *Fenwick Tree*. Suporta diversas operações, sendo bem robusta computacionalmente, porém, possui constantes de alta complexidade e um código verboso.

1.5 Treap

É também uma estrutura dinâmica. Suporta todas as operações básicas possíveis em intervalos, contudo, possui constantes de alta complexidade pois se trata de uma árvore estatística, possui também um código verboso.

2 Objetivos

Objetivo geral: A implementação das estruturas de dados afim de se tornar uma biblioteca para ser utilizada em competições de programação e afins.

Objetivos específicos:

- Realizar uma revisão bibliográfica, abrangendo a literatura em temas relacionados;
- Fundamentar o conceito de notação assintótica;
- Fundamentar as estruturas de dados com suas notações assintóticas e provas de corretude de suas operações;
- Elencar problemas relacionados a cada uma das estruturas abordadas
- Implementar o código de cada uma das estruturas abordadas

3 Metodologia

O desenvolvimento do trabalho se dará pela fundamentação teórica do tema e a implementação de cada uma das estruturas. Entre as atividades propostas, destacam-se os seguintes tópicos:

1. Análise Assintótica
2. Operações frequentes em intervalos
3. Fundamentar os conceitos de *Prefix Sum*
4. Fundamentar os conceitos de *Sparse Table*
5. Fundamentar os conceitos de *Fenwick Tree*
6. Fundamentar os conceitos de *Segment Tree*
7. Fundamentar os conceitos de *Treap*
8. Elencar problemas relacionados a cada uma das estruturas fundamentadas
9. Implementação do código de cada uma das estruturas fundamentadas

4 Cronograma proposto

Etapas	2019											
	Jan	Fev	Mar	Abr	Maio	Jun	Jul	Ago	Set	Out	Nov	Dez
1			x x									
2			x x									
3				x x								
4					x x	x x						
5							x x	x x				
6									x x	x		
7											x x	x
8				x x	x x	x x	x x	x x	x x	x x	x x	
9				x x	x x	x x	x x	x x	x x	x x	x x	

5 Linha e Grupo de Pesquisa

Este trabalho de conclusão de curso se enquadra na linha de pesquisa de fundamentações teóricas de estruturas de dados e complexidade de algoritmos, que pertence ao Grupo de Pesquisa de Fundamentos da Computação (Função).

6 Forma de Acompanhamento/Orientação

O acompanhamento das atividades desenvolvidas será através de encontros virtuais semanais ou quinzenais (de uma hora ou mais), além de troca de *e-mails* entre os encontros caso seja necessário. A adição de novos encontros pode vir a ser necessária de acordo com o desenvolvimento do trabalho. Os artefatos produzidos pelo orientado serão disponibilizados à orientadora em ambientes de acesso mútuo para acompanhamento contínuo.

Referências

- COUTO, Y. S. **Algoritmos em sequências**. 2016. Trabalho de Conclusão de Curso (Graduação), USP (Universidade de São Paulo), São Paulo, Brasil.
- DUARTE, G. L. **Técnicas De Otimização De Programação Dinâmica**. 2017. Trabalho de Conclusão de Curso (Graduação), UNIFESO (Centro Universitário Serra dos Órgãos), Teresópolis, Brasil.
- HALIM, S. et al. **Competitive Programming 3**. [S.l.]: Lulu Independent Publish, 2013. 4-5 p.
- KAWAKAMI, M. M. **Algoritmos em redes de fluxo e aplicações**. 2017. Trabalho de Conclusão de Curso (Graduação), USP (Universidade de São Paulo), São Paulo, Brasil.
- TOMMASINI, S. **Programação Dinâmica**. 2014. Trabalho de Conclusão de Curso (Graduação), USP (Universidade de São Paulo), São Paulo, Brasil.

VERHOEFF, T. 20 years of ioi competition tasks. **Olympiads in Informatics**, v. 3, p. 149–166, 2009.

Karina Girardi Roggia
Orientadora

Adilson Luis Jonck Junior