

## **Proposta do Trabalho de Conclusão de Curso**

Um estudo sobre estrutura de dados retroativas

**Aluna: Beatriz Figueiredo Marouelli**

**Orientadora: Cristina Gomes Fernandes**



Universidade de São Paulo

Instituto de Matemática e Estatística

## Resumo

A classe de estruturas de dados retroativas introduz uma maneira de controlar o histórico de operações realizadas sobre determinada estrutura de dados, modificando sua implementação tradicional para incluir a associação de cada operação a um instante de tempo e a possibilidade de incluir ou excluir operações no passado. O primeiro estudo sobre esse paradigma foi produzido por Demaine, Iacono e Langerman [1], que elaboraram um artigo apresentando a definição formal de retroatividade, uma extensa demonstração sobre a não existência de um método genérico para converter qualquer estrutura em uma versão retroativa, e provaram que as seguintes estruturas possuem versão retroativa eficiente: fila, fila dupla, union-find e fila de prioridades. Este trabalho pretende produzir um texto introdutório para apresentar esse tipo de estrutura e as técnicas envolvidas.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>3</b>
1.1	Retroatividade Parcial . . . . .	3
1.2	Retroatividade Total . . . . .	3
<b>2</b>	<b>Objetivos</b>	<b>4</b>
<b>3</b>	<b>Metodologia</b>	<b>5</b>
<b>4</b>	<b>Plano de trabalho e cronograma</b>	<b>6</b>
	<b>Referências</b>	<b>7</b>

# 1 Introdução

Um problema estrutural pode ser facilmente rearranjado sob a perspectiva retroativa. No geral, uma estrutura de dados recebe operações de modificação (*update*) e operações de consulta, que são executadas obedecendo uma certa sequência.

Assumindo que apenas uma operação é feita a cada instante de tempo, tem-se a lista  $U = [u(t_1), \dots, u(t_m)]$  de *updates* da estrutura, onde  $u(t_i)$  é uma operação de modificação efetuada no instante  $t_i$  e  $t_1 < t_2 < \dots < t_m$ .

Normalmente, numa estrutura de dados tradicional, as operações são executadas na ordem em que são dadas. Isso equivale a dizer, utilizando a notação apresentada, que a lista  $U$  sofre apenas inserções e sempre no extremo final.

## 1.1 Retroatividade Parcial

Uma estrutura é dita parcialmente retroativa se ela permite que na lista  $U$  sejam inseridas ou deletadas operações de modificação em qualquer instante de tempo, ou seja, possivelmente no passado, e não apenas inserções no presente como nas estruturas usuais. As operações de consulta continuam sendo realizadas apenas no presente.

## 1.2 Retroatividade Total

Na definição anterior pode-se observar que, como a própria nomenclatura insinua, o controle do histórico é parcial. As operações permitem alterar o passado da estrutura, mas não observá-lo diretamente. Quando uma estrutura é totalmente retroativa, além de conseguir deletar ou inserir operações no passado, também é possível fazer consultas sobre o estado da estrutura em qualquer instante de tempo, não apenas no presente.

## 2 Objetivos

A finalidade deste trabalho de conclusão é estudar boa parte do artigo de Demaine *et al.* [1], implementando várias das estruturas de dados parcial e totalmente retroativas lá expostas, e redigindo um texto didático para apresentar esse tema e as estruturas de dados retroativas estudadas. Para exemplificar o uso da retroatividade, também é parte desta proposta pesquisar sobre aplicações dessas estruturas e escolher uma para ser estudada a fundo e implementada.

### 3 Metodologia

O foco do estudo será no artigo de Demaine *et al.* [1], mas outros quatro artigos que abordam retroatividade, e que são interessantes para este trabalho, foram encontrados durante a pesquisa preliminar.

Os dois primeiros artigos deste levantamento desenvolvem estruturas retroativas para aplicá-las no contexto de geometria computacional: Dickerson M.T., Eppstein D. e Goodrich M.T. [2] apresentam a clonagem de diagramas de Voronoi através de estruturas retroativas, e Goodrich M.T. e Simons J.A. [3] descrevem uma estrutura retroativa para resolver os problemas *range search* e *nearest neighbor*.

Por fim, os dois últimos artigos deste levantamento são, respectivamente, um complemento do artigo introdutório de retroatividade e uma aplicação decorrente desse complemento: Demaine *et al.* [4] implementam uma fila de prioridades totalmente retroativa e Sunita e Garg D. [5] mostram como dinamizar o algoritmo de Dijkstra usando uma fila de prioridades totalmente retroativa.

Conforme o decorrer das atividades primárias, o material de alguns desses outros artigos poderá ser incluído também.

## 4 Plano de trabalho e cronograma

Para a execução deste trabalho, estão listadas abaixo as principais atividades previstas.

**Atividade 1:** pesquisa preliminar e estudo do artigo introdutório sobre estruturas de dados retroativas;

**Atividade 2:** estudo de estruturas de dados com operações comutativas e inversíveis;

**Atividade 3:** estudo e implementação da fila e da pilha totalmente retroativas;

**Atividade 4:** estudo e implementação da fila dupla totalmente retroativa;

**Atividade 5:** estudo e implementação da fila de prioridade parcialmente retroativa;

**Atividade 6:** estudo e implementação de uma aplicação de estruturas retroativas;

**Atividade 7:** escrita da monografia;

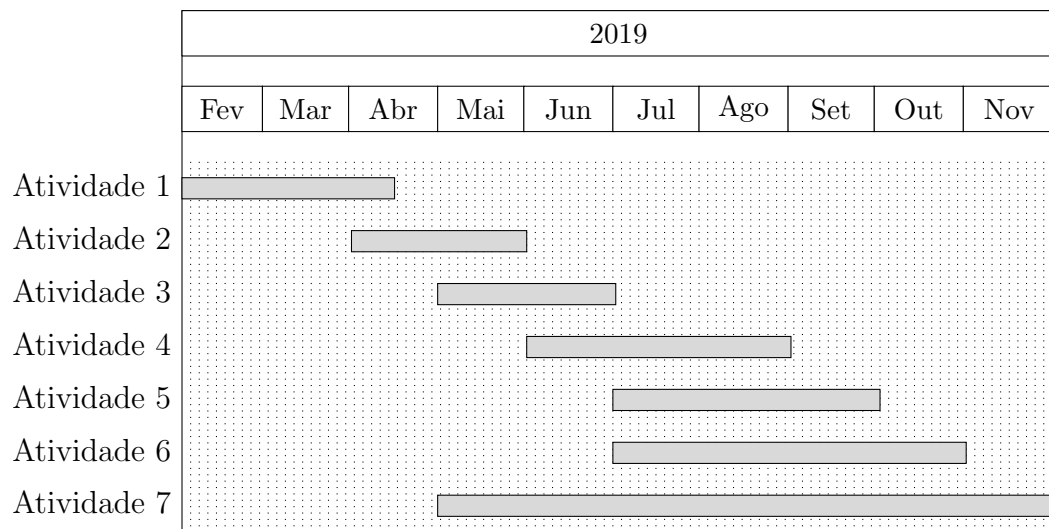


Tabela 1: Diagrama contendo o cronograma.

## Referências

- [1] Demaine E. D., Iacono John, and Langerman Stefan. Retroactive data structures. *ACM Transaction on Algorithms*, Vol. 3(No. 2), May 2007.
- [2] Dickerson M.T., Eppstein D., and Goodrich M.T. Cloning voronoi diagrams via retroactive data structures. *de Berg M., Meyer U. (eds) Algorithms – ESA 2010*, Vol. 6346, 2010.
- [3] Goodrich M.T. and Simons J.A. Fully retroactive approximate range and nearest neighbor searching. *ISAAC 2011. Lecture Notes in Computer Science, vol 7074. Springer, Berlin, Heidelberg*, 2011.
- [4] Demaine E.D., Kaler T., Liu Q., Sidford A., and Yedidia A. Polylogarithmic fully retroactive priority queues via hierarchical checkpointing. *Dehne F., Sack JR., Stege U. (eds) Algorithms and Data Structures. WADS 2015. Lecture Notes in Computer Science. Springer, Cham*, Vol. 9214, 2015.
- [5] Sunita and Garg D. Dynamizing dijkstra: A solution to dynamic shortest path problem through retroactive priority queue. *Journal of Kind Saud University - Computer and Information Sciences*, 2018.