

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

Relatório Final de MAC0215
Atividade Curricular em Pesquisa

Estruturas de Dados Persistentes e Retroativas

Bruno Armond Braga

Orientadora: Prof.^a Dr.^a Cristina Gomes
Fernandes

São Paulo
Dezembro de 2023

Resumo

Uma estrutura de dados (ED) convencional, em geral, oferece suporte a operações de consulta e de modificação. Normalmente, ao sofrer uma modificação, tal ED é atualizada de maneira que perdemos o estado anterior dela. Desse modo, temos acesso apenas ao estado atual da ED.

As chamadas estruturas de dados persistentes e retroativas visam dar ao usuário suporte às operações em todas as versões da ED. Elas têm aplicações em diversos contextos, como por exemplo, em sistemas de bancos de dados, em linguagens funcionais, em sistemas de controle de versão, etc.

O conceito de persistência foi formalmente introduzido em 1986 por Driscoll, Sarnak, Sleator e Tarjan enquanto que a ideia de retroatividade foi introduzida por Demaine, Iacono e Langerman em 2007. Em ambos os casos, o usuário pode fazer consultas a uma versão da ED do passado, de modo que cada versão da ED é vinculada a um instante de tempo. O que distingue esses dois tipos de EDs é a forma de lidar com uma modificação aplicada a uma versão da ED do passado.

Em uma ED persistente, uma alteração num instante t do passado causa uma bifurcação, iniciando um novo ramo do estado da ED a partir daquele instante t . Consultas a versões anteriores permanecem dando as mesmas respostas, e várias versões “paralelas” da ED coexistem.

Já em retroatividade, uma alteração da ED num instante t do passado repercute no estado da ED em todos os instantes a partir de t . Consultas à ED em um instante posterior a t podem passar a dar respostas distintas das respostas obtidas antes desta alteração ter ocorrido.

Resultados Alcançados

Com essa pesquisa, fica claro que as estruturas de dados temporais são bastante sofisticadas, mas oferecem operações impressionantes.

Ao longo do estudo, foi possível explorar e implementar as seguintes estruturas:

- **Retroatividade**

- Implementação de uma pilha com as operações: insert, push, insert pop, delete push, delete pop, size, top, k-th e print (Cap. 3, [1]).

- **Persistência**

- Implementação de uma pilha com as operações: push, pop, size, top e com a operação k-th consumindo tempo linear no número de elementos na pilha (Cap. 3, [3]);
- Implementação de uma pilha alternativa com a operação k-th utilizando contadores binários consumindo tempo logarítmico (Cap. 1, [3]);
- Implementação de mais uma pilha alternativa, esta com a operação k-th utilizando a representação skew-binary, também com consumo de tempo logarítmico (Cap. 2, [3]);
- Implementação de uma fila com as operações: enqueue, dequeue, size, first, last e k-th; (Cap. 3, [3])
- Implementação de uma treap funcional;
- Implementação de uma árvore binária de busca funcional;
- Implementação de uma deque com ancestral de nível (LA) e primeiro ancestral comum (LCA) (Cap. 4, [3]);
- Implementação de uma deque recursiva proposta por Kaplan (Cap. 5, [3]);
- Implementação de uma deque mais eficiente proposta por Kaplan e Tarjan, com todas as operações com custo $O(1)$ com exceção do k-th com custo logarítmico (Cap. 6, [3]).

Vale ressaltar que a implementação de todas essas estruturas pode ser encontrada no repositório do Github disponibilizado no início da pesquisa:

<https://github.com/BrunoArmondBraga/EstruturasTemporais>

Histórico das Atividades Realizadas

- Agosto (4 semanas): 22h
 - Estudo inicial sobre estruturas temporais - 5 horas.
 - Estudo e implementação de três pilhas persistentes diferentes. A primeira com a operação k-th linear, outra com k-th usando ponteiros binários e uma última usando jump pointers - 10 horas.
 - Estudo e implementação de uma fila persistente - 4 horas.
 - Estudo inicial de deque persistentes - 3 horas.
- Setembro (4 semanas): 23h
 - Implementação de uma treap persistente usando ancestral de nível e primeiro ancestral comum - 8 horas.
 - Estudo e implementação de uma árvore binária de busca funcional - 7 horas.
 - Estudo e implementação de uma pilha retroativa - 8 horas.
- Outubro (4 semanas): 23h
 - Estudo da deque persistente recursiva proposta por Kaplan - 10 horas.
 - Implementação da deque persistente recursiva proposta por Kaplan com void pointers - 13 horas.
- Novembro (5 semanas): 26h
 - Estudo da deque persistente proposta por Kaplan e Tarjan - 11 horas.
 - Implementação da deque persistente proposta por Kaplan e Tarjan com void pointers - 15 horas.
- Dezembro (1 semana): 11h
 - Desenvolvimento da operação k-th na deque persistente de Kaplan e Tarjan - 5 horas.
 - Finalizações, organização final do repositório no Github e redação do relatório - 6 horas.

Referências

- [1] Beatriz Figueiredo Marouelli, *Um estudo sobre estruturas de dados retroativas*, Trabalho de Conclusão de Curso, BCC-IME-USP, 2019. Disponível em: <https://bccdev.ime.usp.br/tccs/2019/bfm/>.
- [2] Cristina Gomes Fernandes, Notas de aulas de MAC0385 - Estruturas de Dados Avançadas, 2023. Disponível em: https://www.ime.usp.br/~cris/aulas/23_2_6957.
- [3] Yan Soares Couto, *Estruturas de dados persistentes*, Dissertação de Mestrado, IME-USP, 2018. Disponível em: <https://www.teses.usp.br/teses/disponiveis/45/45134/tde-24092019-181655/pt-br.php>.