O problema da conectividade dinâmica em grafos

Gabriel de Russo e Carmo

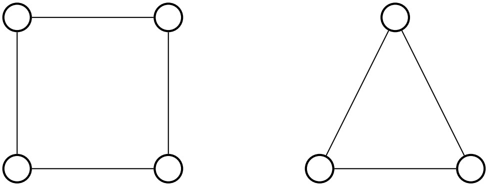
Universidade de São Paulo

gabrielrcarmo@gmail.com

Novembro de 2018

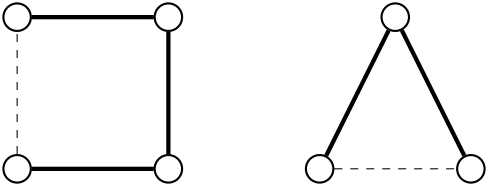
Grafos

Um grafo é um par ordenado (V, A). Cada elemento de A é um par não-ordenado de elementos de V. Os elementos de V são chamados de **vértices**. Os elementos de A são chamados de **arestas**.



Florestas geradoras

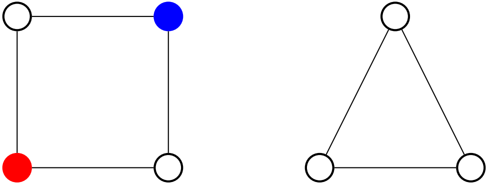
Uma **floresta geradora** é um subgrafo que contém uma árvore geradora de cada um de seus componentes. Equivalentemente, uma floresta geradora é um subgrafo acíclico maximal.



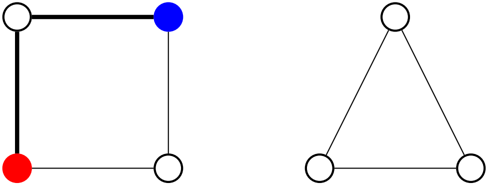
Conectividade dinâmica

Queremos manter um grafo sujeito a atualizações e consultas.
Uma **atualização** é uma adição ou remoção de aresta.
Uma **consulta** é uma pergunta "dois vértices estão conectados?".

Os vértices **vermelho** e **azul** estão conectados?

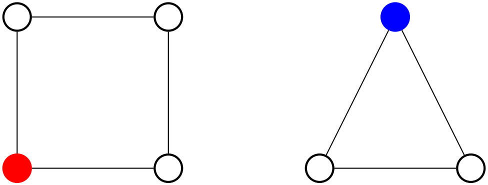


Os vértices **vermelho** e **azul** estão conectados?

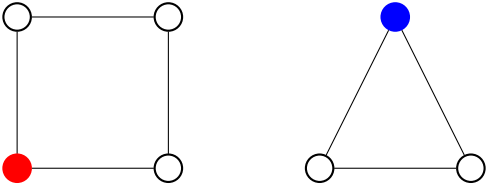


Sim!

Os vértices **vermelho** e **azul** estão conectados?

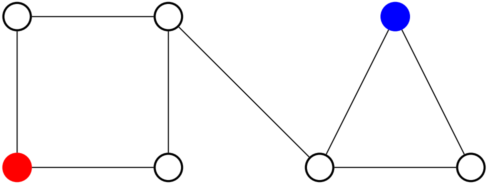


Os vértices **vermelho** e **azul** estão conectados?

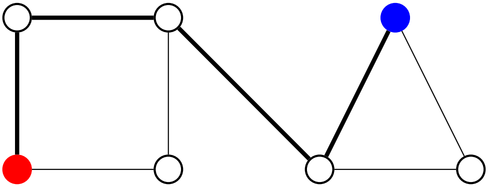


Não!

Após a adição da aresta, os vértices **vermelho** e **azul** estão conectados?

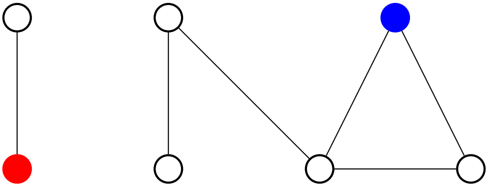


Após a adição da aresta, os vértices **vermelho** e **azul** estão conectados?

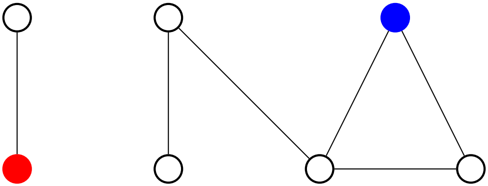


Sim!

Após a remoção das arestas, os vértices **vermelho** e **azul** estão conectados?



Após a remoção das arestas, os vértices **vermelho** e **azul** estão conectados?



Não!

Estrutura de HDT

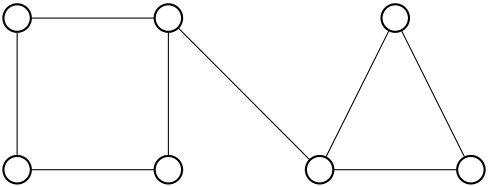
A solução estudada mantém diversas florestas geradoras do grafo [1]. Com isso, conseguimos responder consultas em $O(\lg n)$ e realizar atualizações em $O(\lg^2 n)$ amortizado.

Estrutura de HDT

A solução estudada mantém diversas florestas geradoras do grafo [1]. Com isso, conseguimos responder consultas em $O(\lg n)$ e realizar atualizações em $O(\lg^2 n)$ amortizado.
A ideia é que uma floresta geradora contém informação suficiente para responder as consultas de conectividade.

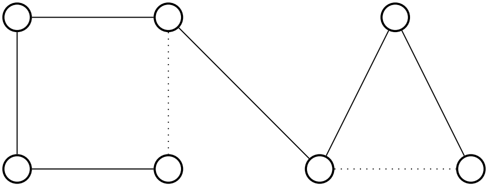
Estrutura de HDT

A solução estudada mantém diversas florestas geradoras do grafo [1]. Com isso, conseguimos responder consultas em $O(\lg n)$ e realizar atualizações em $O(\lg^2 n)$ amortizado.
A ideia é que uma floresta geradora contém informação suficiente para responder as consultas de conectividade.



Estrutura de HDT

A solução estudada mantém diversas florestas geradoras do grafo [1]. Com isso, conseguimos responder consultas em $O(\lg n)$ e realizar atualizações em $O(\lg^2 n)$ amortizado.
A ideia é que uma floresta geradora contém informação suficiente para responder as consultas de conectividade.



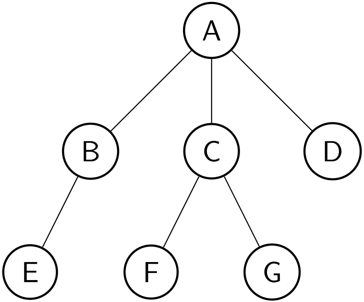
Florestas podem ser representadas de maneira muito eficiente.

Sequências eurelianas

Definimos a **sequência eureliana** [2] de uma árvore T enraizada num vértice r por uma trilha eureliana de T começando em r.

Sequências eurelianas

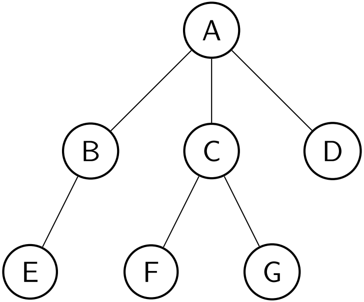
Definimos a **sequência eureliana** [2] de uma árvore T enraizada num vértice r por uma trilha eureliana de T começando em r.



Uma possível sequência eureliana da árvore da figura é
`AA AB BB BE EE EB BA AC CC CF FF FC CG GG GC CA AD DD DA`.

Sequências eurelianas

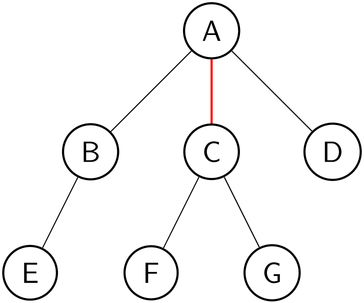
Remover uma aresta de uma árvore tem um comportamento simples sobre sua sequência eureliana.



AA AB BB BE EE EB BA AC CC CF FF FC CG GG GC CA AD DD DA.

Sequências eurelianas

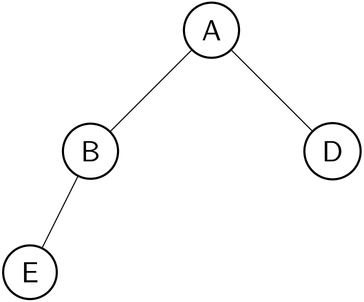
Remover uma aresta de uma árvore tem um comportamento simples sobre sua sequência eureliana.



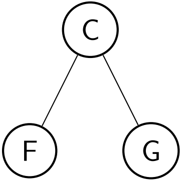
AA AB BB BE EE EB BA AC CC CF FF FC CG GG GC CA AD DD DA.

Sequências eurelianas

Remover uma aresta de uma árvore tem um comportamento simples sobre sua sequência eureliana.



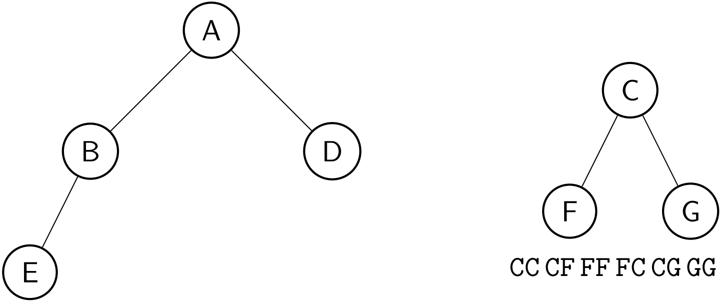
AA AB BB BE EE EB BA AD DD DA.



CC CF FF FC CG GG GC

Sequências eurelianas

Remover uma aresta de uma árvore tem um comportamento simples sobre sua sequência eureliana.



`AA AB BB BE EE EB BA AD DD DA.`

`CC CF FF FC CG GG GC`

Adicionar uma aresta entre duas árvores tem um comportamento similar.

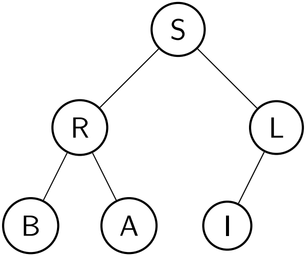
Representação de sequências

Uma sequência pode ser representada por uma árvore de busca binária implícita.

Representação de sequências

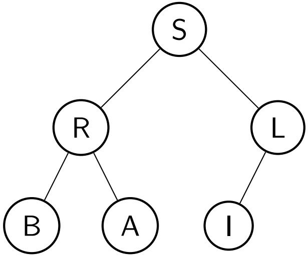
Uma sequência pode ser representada por uma árvore de busca binária implícita.

Considere a sequência `B R A S I L` e uma possível árvore de busca binária implícita que a representa.



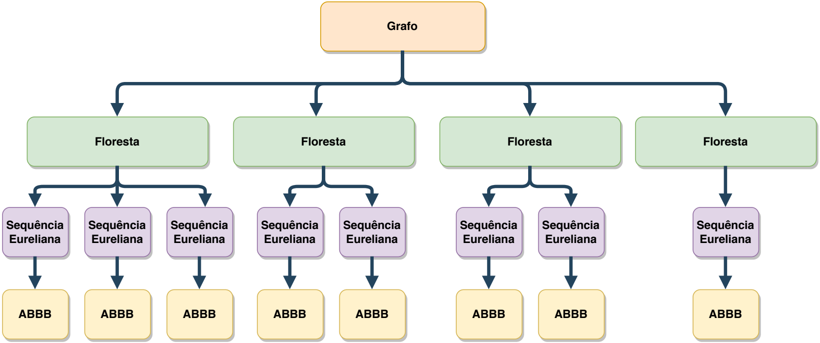
Representação de sequências

Uma sequência pode ser representada por uma árvore de busca binária implícita.
Considere a sequência `B R A S I L` e uma possível árvore de busca binária implícita que a representa.



Usando árvores de busca balanceadas, é possível concatenar e fatiar sequências em tempo $O(\lg n)$ [3].

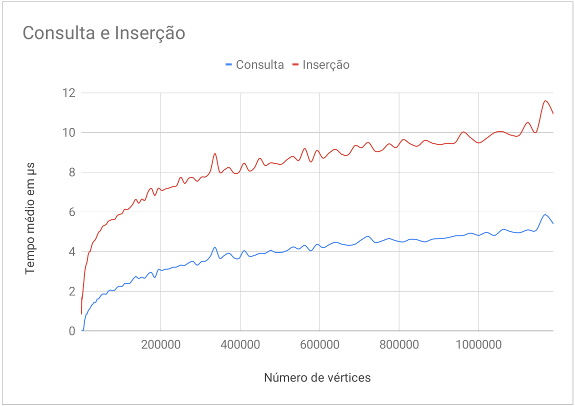
Implementação



A estrutura foi implementada em C++ e o código está disponível em `https://github.com/gabrielrussoc/mac499`.

Desempenho

Testes foram realizados com diversos tamanhos de grafos diferentes.



Referências

Jacob Holm, Kristian de Lichtenberg e Mikkel Thorup (2001)
Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity
J. ACM 48(4), 723—760

Monika Rauch Henzinger e Valerie King (1995)
Randomized dynamic graph algorithms with polylogarithmic time per operation
Proceedings of the Twenty-seventh Annual ACM Symposium on Theory of Computing, STOC '95, 519—527

Robert Endre Tarjan (1983)
Data Structures and Network Algorithms.
Society for Industrial and Applied Mathematics, Philadelphia, PA.

Perguntas?