

Análise de Complexidade - USEI06: Time-Zone Index and Windowed Queries

Resumo

Este documento apresenta a análise detalhada da complexidade algorítmica da implementação da *User Story USEI06 - Time-Zone Index and Windowed Queries*. A solução utiliza um conjunto de **Árvores Binárias de Pesquisa (BSTs) balanceadas**, adaptadas para lidar com **Chaves Duplicadas**, indexadas por `timeZoneGroup`, `latitude` e `longitude`. A complexidade de **busca é O(logN)**, garantindo consultas rápidas e determinísticas em grandes *datasets*.

1. Introdução

A *User Story* USEI06 exige um sistema de indexação robusto para pesquisas rápidas por metadados geográficos e temporais. A solução desenvolvida utiliza BSTs balanceadas para garantir que o tempo de pesquisa de estações por grupo de fuso horário, individualmente ou em janela, seja sempre o mais eficiente possível.

1.1. Características do Dataset

- Aproximadamente 64.000 estações europeias.
 - Chaves de indexação: `timeZoneGroup` (String), `latitude` (Double), `longitude` (Double).
 - Necessidade de preservar estações com a mesma chave (duplicados).
-

2. Arquitetura da Solução

2.1. BSTs Balanceadas e Estrutura dos Dados

A estrutura central é um conjunto de três BSTs implementadas de forma balanceada para limitar a altura da árvore a $O(\log N)$.

Para lidar com **Chaves Duplicadas** (várias estações para o mesmo TZG), cada nó da BST armazena uma **Coleção de Valores** (`List<EuropeanStation>`) em vez de um único objeto.

Estrutura das BSTs

BST	Chave	Tipo de Valor
bstTimeZoneGroup	<code>timeZoneGroup</code> (String)	<code>List<EuropeanStation></code>
bstLatitude	<code>latitude</code> (Double)	<code>List<EuropeanStation></code>
bstLongitude	<code>longitude</code> (Double)	<code>List<EuropeanStation></code>

3. Análise de Complexidade Temporal

3.1. Construção dos Índices (Building Balanced BSTs)

Passo	Complexidade	Justificativa
1. Pré-ordenação da Lista	$O(N \log N)$	Ordenação inicial do <i>dataset</i> pela chave de indexação. Esta é a operação dominante.
2. Construção Recursiva	$O(N)$	A construção da árvore a partir da lista ordenada (usando a mediana) tem custo linear, pois cada elemento é processado uma vez.
Complexidade Total (Build)	$O(N \log N)$	

3.2. Busca por Time Zone Group (TZG)

3.2.1. Busca por Chave Única e Janela

A complexidade combina o tempo de localização da chave na árvore $O(\log N)$ e o tempo de recolha dos resultados $O(M)$.

Cenário	Complexidade	Justificativa
Busca (Chave / Janela)	$O(\log N + M)$	$O(\log N)$ para encontrar a chave (ou o limite do intervalo) na BST balanceada, e $O(M)$ para recolher todos os M resultados.
Pior Caso	$O(N)$	Ocorre se todas as N estações pertencerem ao mesmo TZG ($M=N$).

3.3. Complexidade dos Filtros e Ordenação Final

Operação	Complexidade	Justificativa
Ordenação Final	$O(M \log M)$	Necessária para satisfazer o requisito de ordenação dos M resultados por País e Nome de Estação.
Filtros (ex: <code>filterCountry</code>)	$O(M)$	Filtragem linear nos M resultados, com custo $O(1)$ por estação.

4. Análise de Complexidade Espacial

Componente	Complexidade	Justificativa
Armazenamento da BST	$O(N)$	O espaço é linearmente proporcional ao número de estações, N.
Pilha de Recursão	$O(\log N)$	A profundidade máxima da recursão é logarítmica, devido ao balanceamento da árvore.

5. Conclusão

A implementação da USEI06 atinge a eficiência e escalabilidade esperadas para uma estrutura de dados de pesquisa baseada em chaves, com complexidade de pesquisa $O(\log N + M)$.