Análise de Complexidade Temporal – USEI09

(Proximity Search / NearestnN com Filtros)

Este documento analisa a complexidade temporal do caso de uso USEI09: encontrar as N estações mais próximas de uma coordenada alvo (lat, lon), usando uma KDnTree 2D balanceada, distância Haversine e filtro opcional por grupo de time zone.

Notação e Parâmetros

• S: número total de estações no dataset.

• M: número de nós (buckets) da KDnTree.

• b: tamanho médio de um bucket.

• N: número de vizinhos pedidos.

• K: número de estações avaliadas durante a busca.

• k: número de dimensões (k = 2).

1. Estrutura de dados usada

A KDnTree 2D balanceada possui altura O(log M). Usa-se também um MaxnHeap de tamanho N para manter os melhores candidatos.

2. Custos elementares

2.1 Bucket do nó atual:

Cada estação passa por filtro O(1), distância O(1) e possível atualização do heap O(log N).

Custo por nó: O(|bucket| · log N).

2.2 Decisão de subárvores e poda:

Visita da subárvore oposta somente se |targetCoord - nodeCoord| < maxDistanceInQueue.

3. Complexidade média

Busca visita V = O(log M) nós.

Custo total médio: O(log M + K log N + N log N).

4. Pior caso

Sem poda efetiva: O(M) nós, custo final O(S log N).

5. Filtro de time zone

Custo O(1), reduz K, mas não altera complexidade assintótica.

6. Comparação com abordagem linear

Busca linear: O(S log N).

KD-tree: ~O(log M) no caso médio.

7. Conclusão

A solução é eficiente e adequada para datasets grandes, com caso médio O(log M) e pior caso O(S log N).