

Algoritmos II - Trabalho Prático I

Bruno Buti Ferreira Guilherme
Ricardo Shen

1 Introdução

Neste projeto, foi apresentado o problema de implementar, em Python, um sistema consulta de bares e restaurantes em Belo Horizonte programado, a partir de dados públicos oferecidos pela prefeitura. Neste caso, as buscas são feitas a partir de um sistema interativo a partir do mapa do município, em que o usuário seleciona uma determinada região retangular e recebe de volta todos os estabelecimentos registrados naquela área. Para tal, foi implementada e utilizada uma estrutura de dados conhecida como árvore K-dimensional (Kd Tree), especializada para armazenamento e busca de dados com coordenadas espaciais, de modo a garantir uma execução mais rápida e eficiente do programa

2 Implementação

2.1 Estruturas Auxiliares

Antes de começar programar a árvore, são necessárias algumas estruturas com o objetivo de modularizar e facilitar a implementação da classe principal da Kd Tree.

2.1.1 Info

Representação dos dados importantes do estabelecimento, como nome registrado, nome fantasia e seu endereço

2.1.2 Point

Representação de um estabelecimento no espaço geográfico bidimensional. Vale ressaltar o uso da biblioteca Numpy (np) para otimizar a implementação dessas estruturas

2.1.3 Rectangle

Representação de uma área retangular no espaço, utilizando como base um vértice inferior esquerdo e um vértice superior direito. Apresenta métodos para determinar se um determinado ponto P está contido no retângulo ou se ele intersecciona algum dos eixos de separação dos nós da árvore.

2.1.4 Node

Representação da estrutura fundamental que armazena os dados e a organização hierárquica da árvore. São guardados ponteiros para as subárvores da esquerda e da direita, caso existam, assim como informações do eixo de divisão e o valor da coordenada em que a segmentação foi realizada.

2.2 KdTree

Estrutura de organização espacial dos estabelecimentos, permitindo buscas rápidas a partir de uma região retangular. É armazenado apenas um único nó, a raiz da árvore, a partir de onde todas as inserções e buscas serão realizadas.

2.2.1 Método `build_tree`

Método utilizado para construir a árvore recursivamente a partir de uma lista de pontos. As inserções são iniciadas pela raiz e a cada chamada, a lista de pontos é particionada a partir da sua mediana, utilizada como o ponto armazenado pelo nó atual, com cada sublista sendo passada como parâmetro para a construção das subárvores da direita e da esquerda. As segmentações recursivas são encerradas quando a lista de pontos é totalmente consumida.

2.2.2 Método `search`

Método utilizado para realizar a consulta ortogonal dos pontos baseados em uma área retangular. A busca é feita recursivamente a partir de cada subárvore. Se um determinado nó pertence à região dada, ele é adicionado à uma lista de pontos encontrados. Em seguida, os nós da direita e da esquerda são analisados recursivamente, até que não hajam mais pontos a serem explorados. Diferentemente de uma busca linear por um vetor ou por uma lista encadeada, a estrutura da Kd Tree permite que a consulta seja guiada a partir dos dados geográficos, evitando comparações desnecessárias com pontos claramente fora do escopo fornecido. Essa característica da árvore garante operações de busca rápidas e eficientes, especialmente em situações com milhares de pontos, como neste trabalho

3 Filtragem de dados

Para otimizar o tempo de processamento anterior à inserção dos dados na árvore, foi realizado uma pré-filtragem de todos os estabelecimentos, armazenando em um arquivo CSV apenas os pontos de interesse, bares e restaurantes. Primeiramente, os dados foram baixados diretamente do link fornecido pela prefeitura de Belo Horizonte. A partir de lá, o CSV foi separado em linhas contendo informações dos estabelecimentos e a linha específica contendo a representação de cada coluna. Em seguida, os valores únicos na coluna de *DESCRICAO_CNAE_PRINCIPAL* foram separados e filtrados para manter apenas aqueles que se referem a restaurantes ou bares. Por fim, o dataset original foi novamente filtrado em um novo, contendo apenas os estabelecimentos que se encaixam na descrição, e salvo em um novo arquivo. Todas as operações na árvore são realizadas a partir do CSV contendo apenas os dados já tratados. Ao total, o número de linhas após a manipulação desceu de cerca de 534 mil para menos de 18 mil.

4 Aplicativo Web

O sistema de consultas foi programado utilizando principalmente o framework Dash, designado para a criação de sistemas web interativos. A visualização principal é de um mapa de Belo Horizonte, com a opção de movimentar a visão do mapa e de desenhar uma área retangular, usada para a realização da busca ortogonal de todos os bares e restaurantes contidos naquela determinada área do município, usando como base de pesquisa a árvore Kd demonstrada anteriormente