Introdução

Neste trabalho de programação orientada a objetos foi desenvolvido uma classe chamada TabelaHash que usa o método de hashing para organizar dados. A ideia principal é aplicar essa classe para resolver o problema de dados duplicados em arquivos CSV, que é muito comum em ciência de dados.
O método tradicional de tirar duplicatas normalmente usa ordenação, mas isso tem um custo maior (O(n log n)). Com uma tabela hash, o processo fica mais rápido (O(n)) e pode ajudar bastante quando o arquivo tem muitos registros.

Desenvolvimento

A classe TabelaHash foi feita usando listas dentro de uma lista maior (técnica de encadeamento) para lidar com colisões. Ela tem os métodos:

insert: adiciona uma chave e um dado se a chave ainda não está na tabela.
search: procura uma chave na tabela e retorna o dado.
remove: exclui um elemento pela chave.
__getitem__: permite acessar um índice da tabela.

Também foram criadas duas funções de hash:

hash_simples: transforma a chave em inteiro direto.
hash_dobra: divide a chave no meio e soma as partes para tentar espalhar melhor os valores.

Para testar a classe foi criada a função deduplicar_csv, que lê um arquivo CSV, aplica o método de hash para cada linha e salva só os registros únicos em outro arquivo.
Testes
Foi usado o arquivo people-10000.csv, que tem 10 mil registros com alguns e-mails repetidos. O campo “Email” foi escolhido como chave. O programa percorreu o arquivo e gerou um novo arquivo people-10000-dedup.csv com os dados sem repetição.

Complexidade

Inserção e busca: O(1) em média.
Deduplicação completa: O(n), com n sendo o número de linhas.
Esse resultado mostra que o uso da Tabela Hash é uma boa opção para arquivos grandes.

Dificuldades encontradas

Durante o desenvolvimento apareceram alguns desafios:
Algumas funções de hash davam erro quando a chave não era número, então foi preciso converter para string.
Quando o usuário passa uma coluna errada, o código dava erro, mas isso foi resolvido com uma verificação.
Valores nulos nas chaves também precisaram ser tratados.

Conclusão

O trabalho atingiu o objetivo de implementar uma Tabela Hash funcional e aplicá-la para deduplicação de dados. A função deduplicar_csv facilita bastante porque já cria o nome do arquivo de saída automático se o usuário não passar. O método funciona bem mesmo em arquivos grandes e pode ser útil em outras aplicações além de deduplicação.