Hashing e Busca binária

Relatório Final: Projeto de Avaliação de Estruturas de Dados

---

1. Introdução:

O presente projeto teve como **objetivo avaliar o desempenho de três diferentes estruturas de dados** - busca sequencial, busca binária e hashing - em termos de inserção e pesquisa de registros de pessoas em conjuntos de dados de tamanhos variados.

---

2. Atividades Realizadas:

- Implementação das Estruturas de Dados: Foram implementadas três classes para representar as estruturas de dados de busca sequencial, busca binária e hashing, cada uma com métodos para inserção e pesquisa.
- Geração de Dados: Foi desenvolvida uma função para gerar dados de pessoas aleatórias, incluindo CPF, nome, telefone e senha.
- Testes de Desempenho: Foram realizados testes de inserção e pesquisa para cada estrutura de dados em conjuntos de dados de tamanhos variados (10.000, 100.000 e 1.000.000 de elementos), registrando os tempos médios de execução.

---

3. Resultados Obtidos:

Os resultados obtidos dos testes de desempenho foram os seguintes:

- Inserção:
    - A estrutura de hashing demonstrou ser a mais eficiente em termos de tempo de inserção, especialmente para conjuntos de dados maiores.
    - A busca binária mostrou-se eficiente para inserção em conjuntos de dados já ordenados, mas tornou-se menos eficiente em conjuntos de dados maiores devido à complexidade de tempo de inserção O(n).
- Pesquisa:
    - A busca binária foi a mais eficiente em termos de tempo de pesquisa, especialmente para conjuntos de dados maiores, devido à sua complexidade de tempo- O(log n).
    - A estrutura de hashing também apresentou bom desempenho na pesquisa, especialmente para conjuntos de dados maiores, devido à sua complexidade média de tempo -O(1).

- A busca sequencial foi a menos eficiente em termos de tempo de pesquisa, especialmente para conjuntos de dados maiores, devido à sua complexidade de tempo- O(n).

---

4. Conclusões:

Com base nos resultados obtidos, as seguintes conclusões podem ser alcançadas:

- Para conjuntos de dados de tamanho moderado a grande e operações de pesquisa frequentes, a busca binária e a estrutura de hashing são escolhas eficientes devido ao seu desempenho superior em termos de tempo de pesquisa.
- Para operações de inserção frequentes em conjuntos de dados grandes, a estrutura de hashing é a mais eficiente devido à sua complexidade de tempo de inserção média -O(1).
- A busca sequencial é adequada apenas para conjuntos de dados pequenos ou quando a ordenação dos dados não é uma opção e as operações de pesquisa são menos frequentes.
- No contexto deste **código específico**, que implementa estruturas de dados de busca sequencial, busca binária e hashing, as considerações a serem levadas em conta incluem:

Tamanho do Conjunto de Dados:

- Para conjuntos de dados pequenos, a diferença de desempenho entre as estruturas de dados pode não ser tão significativa. No entanto, para conjuntos de dados maiores, a eficiência das operações de inserção e pesquisa se torna crucial.

Frequência de Inserções e Pesquisas:

- Se o aplicativo envolver principalmente inserções e pesquisas ocasionais em um grande conjunto de dados, a estrutura de hashing pode ser uma escolha eficiente devido à sua complexidade média de tempo-O(1) para operações de inserção e pesquisa.
- Se o aplicativo envolver muitas pesquisas e poucas inserções ou vice-versa, a escolha entre busca sequencial e busca binária pode depender da ordenação dos dados e das características específicas das operações.

Ordenação dos Dados:

- A busca binária requer que os dados estejam ordenados, enquanto a busca sequencial e o hashing não têm essa exigência. Portanto, se os

dados já estiverem ordenados ou se a ordenação for aceitável, a busca binária pode ser uma opção eficiente.
- Se a ordenação dos dados for impraticável ou não desejada, a busca sequencial ou o hashing podem ser mais adequados.

Colisões de Hash:

- Ao usar hashing, é importante considerar a possibilidade de colisões, onde duas chaves diferentes resultam no mesmo índice de hash. O tratamento de colisões, como encadeamento ou resolução de colisões por sondagem, pode afetar o desempenho geral da estrutura de hashing.

Eficiência de Memória:

- Embora o código atual não analise especificamente a eficiência de memória das estruturas de dados, é importante considerar a quantidade de memória utilizada por cada estrutura, especialmente para conjuntos de dados grandes.