Relatório questões teóricas

Aluno

Richard Matheus Bezerra Ataliba (20240023956)

link do repositório

https://github.com/RichardMatheus03/lista-EDB1

Capturas de tela rodando o test_cpp

Questões discursiva - relatório

Questão 6.2 — Impacto da Ordenação nos Algoritmos de Busca

Sim, em muitos casos vale a pena ordenar os dados previamente quando precisamos realizar **várias buscas no mesmo conjunto e também quando esse conjunto é grande fazendo com que a diferença entre os algoritmos de busca se "alargue" mais**. Alguns dos principais pontos são

- **Impacto nas buscas** – Depois de ordenado, podemos utilizar algoritmos mais rápidos, como a **busca binária**, que tem custo **O(log n)** por busca, em vez da busca sequencial que é **O(n)**.
- **Cenário a longo prazo** – Se o conjunto será consultado muitas vezes, o custo da ordenação compensa. Por exemplo:
  - Uma única busca → melhor usar busca sequencial (**não compensa ordenar**).

    - Muitas buscas → ordenar uma vez (**O(n log n)**) e depois fazer várias buscas em **O(log n)** cada. Isso gera ganho de desempenho no sistema como um todo.
- **Relação com análise assintótica** – Na prática, a escolha depende da **quantidade de buscas** em relação ao **tamanho dos dados**:
    - Para poucas buscas, o **custo de O(n log n)** da ordenação é maior do que simplesmente fazer algumas buscas sequenciais.
    - Para muitas buscas, o custo inicial da ordenação se dilui e o tempo total passa a ser muito menor com a busca binária.

Questão 6.3 — Recursão x Iteração

- **Diferença principal**
    - Recursão: a função chama a si mesma até chegar na condição de parada.
    - Iteração: usa laços de repetição (`for`, `while`) para repetir instruções.
- **Prós da recursão**
    - Código mais simples e intuitivo em problemas como árvores, grafos ou definições matemáticas (fatorial, Fibonacci).
    - Facilita o raciocínio em problemas de divisão e conquista.
- **Contras da recursão**
    - Gera mais consumo de memória, pois cada chamada ocupa espaço na pilha.
    - Pode causar estouro de pilha em chamadas muito profundas.
    - Normalmente mais lenta que a iteração devido ao overhead das chamadas.
- **Prós da iteração**
    - Mais eficiente em tempo e memória.
    - Evita problemas de limite de chamadas.
    - Melhor para laços simples e repetitivos.
- **Contras da iteração**

    - Código pode ficar menos legível em problemas naturalmente recursivos.
- **Resumo**
    - Recursão é ideal quando o problema se descreve naturalmente de forma recursiva, como árvores e algoritmos de ordenação.
    - Iteração é preferível quando eficiência e consumo de memória são prioridades.

Questão 6.4 — Análise de um Cenário Real

- Solução que eu recomendaria
    - Busca sequencial (linear), pois os dados chegam desordenados e precisam ser tratados imediatamente.
- **Justificativa técnica**
    - Pensando no contexto de uma empresa de logistica, esses pacotes não seria muito mais do que milhares por dia, o que não seria uma quantidade suficientemente grande pra fazer valer a pena um pre-processamento de ordenação
    - Busca sequencial tem custo **O(n)**, mas funciona direto em dados não ordenados.
    - Tempo de resposta é imediato, sem necessidade de pré-processamento.
- **Caso houvesse pré-processamento**
    - Seria possível ordenar os códigos com um algoritmo de ordenação eficiente (**O(n log n)**).
    - Depois da ordenação, a busca binária seria a melhor opção (**O(log n)** por busca).
    - Isso traria ganho significativo em cenários com muitas consultas sobre o mesmo conjunto de dados.