UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
INSTITUTO METRÓPOLE DIGITAL
IMD0029 - ESTRUTURA DE DADOS BÁSICAS 1
PROF.º JOÃO GUILHERME

RELATÓRIO DAS QUESTÕES TEÓRICAS

SABRINA DA SILVA BARBOSA

NATAL
2025

Questão 6.1 — Complexidade de Algoritmos de Busca

Tema: Busca sequencial vs. busca binária

Enunciado:

Compare os algoritmos de busca sequencial e busca binária em termos de complexidade, aplicabilidade e limitações. **Discuta**:

- **Em quais contextos cada um é mais apropriado?**

A busca Sequencial, é mais adequada para a realização de busca em um volume pequeno de dados, pois em determinadas situações acaba sendo necessário percorrer todo o vetor de dados para encontrar o alvo/chave desejado. Nesse sentido, se o alvo está na última posição e o vetor de dados é muito grande, acabaria demorando muito para encontrar o alvo e, no pior dos casos, se o dado não está contido nesse vetor e ele é muito grande, iria demorar para informar ao usuário/programador que o dado-alvo não está lá. No melhor dos casos seria quando o alvo está na primeira posição. Uma vantagem é que esse tipo de busca, ou seja, a busca Sequencial pode ser realizada em vetores com dados desordenados.

Não obstante, em relação à busca Binária, é mais adequada para a realização de busca em um volume muito grande de dados, pois usa a divisão em 2 partes do vetor descartando uma das partes – dividir para conquistar, comparando o alvo/chave indicado se o dado estaria no meio, à esquerda ou à direita desse meio da divisão, sendo isso feito à cada divisão, evitando que fosse necessário percorrer todo o vetor de dados para buscar o alvo/chave desejado. Nesse sentido, se o alvo está na última posição e ele é um vetor muito grande, serão feitas muitas divisões, porém encontra o dado-alvo com muito mais rapidez em relação à busca Sequencial para a mesma quantidade de dados, mesmo tamanho de vetor, o mesmo ocorre quando o dado-alvo não está contido no vetor de dados, apesar de demorar um pouco, essa demora é muito menor quando se compara à busca Sequencial. No pior dos casos é quando o vetor está desordenado, ou seja, sendo impossível de realizar a busca Binária como está, é necessário realizar uma ordenação antes de realizar a busca e, quando o volume de dados é muito grande, acaba sendo custoso para realizar essa ordenação, pois precisa percorrer todo o vetor realocando os dados. Dessa forma, é imprescindível fazer a ordenação em vetores desordenados para realizar a busca Binária. No melhor dos

casos é quando o alvo é o meio da divisão. Uma vantagem, não verifica todo o vetor de dados para buscar o alvo desejado.

- **Quais são os pré-requisitos para aplicar a busca binária corretamente?**

Os dados precisam estar ordenados (crescente ou decrescente).

- **Como a ordenação influencia a escolha do algoritmo de busca?**

Como a busca Sequencial obrigatoriamente percorre todo o vetor de dados de forma sequencial, não faz diferença se os dados estão ou não ordenados, pois irá passar sequencialmente por eles até encontrar o alvo. Nesse sentido, se você recebe um vetor desordenado para realizar uma busca de um dado nesse vetor, é mais interessante usar o método/algoritmo da busca Sequencial. Entretanto, se os dados já estão organizados de forma crescente ou decrescente, o algoritmo da busca binária acaba sendo mais indicado para ser usado, pois não irá percorrer todo o vetor para realizar a busca do dado-alvo.

Exemplo.:

Dado um vetor de 10 posições e que se quer o dado 2

vector<int> arr1 = {7, 5, 9, 3, 1, 4, 0, 2, 8, 6};

Ao realizar a busca sequencial nesse vetor, vai passar por cada uma das posições até encontrar o 2 na posição 7 e retornar à posição do alvo.

Todavia, para a busca binária seria necessário primeiro ordenar o vetor, sendo um custo, para então realizar a busca, fazendo ainda algumas divisões até encontrar o 2:

Ordenado - vector<int> arr1 = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};

1ª divisão – {0, 1, 2, 3, 4} – descarta o meio e a parte direita

2ª divisão – não descarta o meio, pois encontrou o alvo na posição 2 que é o meio da divisão.

Questão 6.2 — Impacto da Ordenação nos Algoritmos de Busca

Tema: Pré-processamento e desempenho de algoritmos

Enunciado: Considere uma situação onde é necessário fazer várias buscas sobre um mesmo conjunto de dados.

● **Vale a pena ordenar os dados previamente?**

Depende da quantidade de dados. Se for uma quantidade pequena, utilizar uma busca sequencial é o suficiente, pois é possível de realizar a busca de forma desordenada e não haveria um gasto de recurso e tempo para realizar a ordenação. Porém, se estamos tratando de um grande volume de dados, o mais interessante é de fato gastar um pouco de recurso e tempo realizando a ordenação e usufruir do algoritmo de busca binária, pois para um grande volume de dados a busca sequencial acaba sendo menos efetiva, pois corre o risco passar muito tempo procurando um alvo já que faz isso de forma sequencial, mesmo os dados estando desordenados é um risco alto de ser demoroso que não vale a pena arriscar, principalmente por serem realizadas várias buscas naquele mesmo conjunto de dados.

● **Qual o impacto dessa escolha no desempenho do sistema a longo prazo?**

Se for uma quantidade pequena de dados praticamente não haverá diferenças significativas, em termos de desempenho, para ordenar ou realizar a busca sequencial, pois apesar de ocorrer uma leve demora para realizar algumas buscas, não terá um impacto negativo, dado que ambos os algoritmos de busca estariam gastando quase o mesmo tempo, ainda que a busca binária apresente melhor desempenho, a diferença é de apenas alguns minutos ou menos que isso e isso considerando que não estamos tratando de um sistema de alto risco, como o monitoramento dos batimentos cardíacos de uma pessoa em que cada minuto precisa ser considerado rigidamente. Entretanto, se o volume de dados é alto, ao longo do tempo, isso poderá ser desastroso, especialmente se levarmos em consideração a própria arquitetura do hardware, podendo haver, dentre outras coisas, o superaquecimento ou estouro de memória, que podem levar o sistema a "crashar", sendo necessário reiniciar a máquina, havendo a perda de algumas operações que não foram possíveis de serem salvas a tempo na memória. Nesse sentido, apesar de ainda ter o risco de situações como essas ocorrerem ao realizar uma ordenação, fazer isso uma única vez e realizar a busca binária acaba sendo mais seguro, pois esse tipo de busca reduz o conjunto de dados

pela metade a cada verificação, diminuindo o uso da quantidade de recursos disponíveis, ao contrário do algoritmo de busca sequencial que permanece constante.

● **Como isso se relaciona com o tempo de execução dos algoritmos envolvidos?**

Como a busca sequencial precisa percorrer dado a dado, um após o outro, dependendo da quantidade de dados e do alvo a ser buscado, esse algoritmo pode ser muito rápido ou muito demoroso, porém ainda semelhante em relação ao algoritmo de busca binária se tratando de um mesmo conjunto pequeno de dados. Não obstante, a quantidade de dados a serem percorridos é alta o risco de demora para a busca sequencial aumenta consideravelmente, tornando-se inviável de ser utilizada para esse tipo de situação, ou seja, recomenda-se o uso da busca binária, ainda que seja necessário realizar uma ordenação antes, se for o caso de realizar várias buscas num mesmo conjunto de dados, apesar de ter alguns riscos ainda seria a melhor opção, pois a longo prazo o tempo de busca seria muito menor (considerável) quando comparado à busca sequencial.

Questão 6.3 — Recursão x Iteração

Tema: Estratégias de resolução de problemas

Enunciado: Explique as principais diferenças entre recursão e iteração na resolução de problemas algorítmicos. Discuta:

● **Quais são os prós e contras de cada abordagem?**

A recursão é uma forma de lidar com problemas computacionais que diminui a quantidade de linhas de código, sendo mais simples de ser usada a depender do problema. Também usa da ideia de dividir para conquistar, pois divide os problemas em subproblemas menores. Entretanto, ela pode acabar utilizando muita memória, pois ao chamar a si mesma acaba realizando um empilhamento de dados e, a depender da quantidade de dados a serem usados, pode gerar um problema de "overflow" na memória (estouro da memória).

Já em relação à iteração, geralmente acaba sendo mais eficiente em termos de uso de memória, pois não realiza um empilhamento de dados e evita sobrecarga de chamadas de função, entretanto, acaba utilizando mais linhas de código deixando mais longo e até mesmo podendo dificultar o entendimento para algumas pessoas já que usa de mais "informações" na construção desse método de resolução de problemas, especialmente quando se trata de problemas naturalmente recursivos.

● **Em quais tipos de problemas a recursão pode ser mais vantajosa?**

Problemas que possuem estrutura naturalmente recursiva, como o processamento de estruturas de árvore e aqueles tipos de problema que têm como base a noção do "dividir para conquistar".

● **Existe alguma desvantagem de desempenho ou consumo de memória?**

Para a recursão, temos que ela costuma ser menos eficiente quando existe muitas chamadas, pois pode levar a um estouro de pilha "stack overflow", ou seja, isso acaba sendo uma desvantagem dado que cada chamada recursiva ocupa espaço de memória na pilha de chamadas do sistema. Nesse sentido, a Iteração, acaba sendo normalmente mais eficiente em termos de tempo e memória na maioria dos casos, pois evita essa sobrecarga de chamadas.

Inclua um exemplo simples em pseudocódigo ou linguagem C/C++ para ilustrar sua resposta.

Questão 6.4 — Análise de um Cenário Real

Tema: Escolha de algoritmos na prática

Enunciado: Imagine que você trabalha no setor de tecnologia de uma empresa de logística. O sistema precisa localizar rapidamente pacotes com base em códigos de rastreio que chegam em tempo real. Os dados chegam desordenados.

● **Qual algoritmo de busca você recomendaria para este sistema?**

Busca sequencial.

● **Justifique tecnicamente sua escolha considerando tempo de resposta, necessidade de ordenação e custo computacional.**

Os dados chegam em tempo real, porém a depender da situação não são todos os dias que há uma grande quantidade de dados a serem verificados, nesse sentido os dados que chegam não são sempre a mesma quantidade, podendo ser grande ou não. Nesse caso, por não haver uma certeza acerca da quantidade de dados que chegam, e podendo haver muitas situações em que os dados que chegam são poucos, a busca sequencial acaba sendo a melhor opção, pois não é necessário ter o tempo de ordenar para depois realizar a busca, ou seja, como a complexidade da busca linear, sendo o caso da busca sequencial, é O(n), onde n é o número de pacotes, acaba não exigindo memória ou tempo adicionais para ordenação. Nesse sentido, acaba sendo mais barata computacionalmente no início da operação, enquanto os dados ainda estão sendo recebidos ou processados. Salientando ainda que para poder usar a busca binária seria necessário já ter todos os dados prontos para então realizar uma ordenação, o que não é o caso, pois estão chegando em tempo real, sendo mais um motivo para não usar a busca binária nessa situação.

● **Caso os dados pudessem ser pré-processados, sua escolha mudaria?**

Sim, pois como os dados já estão pré-processados, se tem o tamanho exato de cada conjunto, mesmo que para um conjunto pequeno de dados não faça muita diferença usar ou não a busca sequencial ou binária e usufruir da ordenação, a busca binária é mais rápida do que a busca sequencial, ainda que tenha o custo da ordenação, dado que a complexidade da busca binária é de complexidade O(log n), ou seja, não é linear como a busca sequencial, sendo cada vez melhor ao longo do tempo e do tamanho do conjunto de dados.