Atividade B3-2 – Comparando Eficiência de Algoritmos

Enunciado:

Instruções

Vimos que o tempo de execução de um algoritmo é a quantidade de passos básicos para tratar uma entrada de n elementos ***t(n).***

Com base nesta premissa, elabore uma tabela contagem de tempo, comparando os três algoritmos 1)Busca Linear, 2) Busca Linear em ordem e 3) Busca Binária, que sequem anexo.

Demonstre a análise de cada algorimo para justificar cada linha produzida na tabela

A tabela deverá ter a seguinte configuração:

	BuscaLinear	BuscaLinearEmOrdem	BuscaBinaria
$x \in A$			
$x = A[1]$			
$x = A[n]$			
$x \notin A$			

Resolução:

Vamos analisar cada um dos três algoritmos fornecidos: Busca Linear, Busca Linear em Ordem e Busca Binária. A análise será feita considerando o número de passos básicos necessários para cada um dos casos:

1. Busca Linear (A, n, x): Percorre todos os elementos da lista A até encontrar o elemento “x” ou até percorrer toda a lista.

2. Busca Linear em Ordem (A, n, x): Percorre a lista de forma linear, mas interrompe a busca assim que encontra um elemento maior que “x”, assumindo que a lista está ordenada.

3. Busca Binária (A, n, x): Divide a lista pela metade a cada iteração, buscando “x” em uma lista ordenada.

Vamos analisar cada cenário (coluna da tabela) para cada algoritmo:

Cenário 1: X ∈ A

- Busca Linear:

  - A busca linear vai iterar até encontrar “x” e, em média, espera-se que o elemento seja encontrado na metade da lista. Assim, o número de passos é “n/2” no caso médio.

- Busca Linear em Ordem:

  - Mesmo que a lista esteja ordenada, a busca linear em ordem tem o mesmo comportamento que a busca linear, pois não há ganho se “x” for um elemento pequeno ou grande dentro da lista, resultando em um número médio de passos de “n/2”.

- Busca Binária:

  - A busca binária divide o problema pela metade a cada iteração. O tempo de execução é log2(n), pois é o número de vezes que podemos dividir “n” por 2 até chegar a 1.

Cenário 2: X = A[1]

- Busca Linear:

  - O melhor caso ocorre quando “x” é o primeiro elemento da lista. Apenas 1 passo é necessário.

- Busca Linear em Ordem:

  - Assim como na busca linear, apenas 1 passo é necessário para encontrar “x” no primeiro elemento.

- Busca Binária:

  - A busca binária, mesmo no melhor caso, ainda precisa calcular o índice médio e fazer comparações, resultando em um tempo de execução constante, “O(1)”.

Cenário 3: X = A[n]

- Busca Linear:

  - O pior caso para a busca linear ocorre quando “x” é o último elemento da lista. Neste caso, a busca linear precisa percorrer todos os “n” elementos.

- Busca Linear em Ordem:

- Mesmo que a lista esteja ordenada, a busca linear em ordem ainda precisa percorrer todos os “n” elementos até encontrar “x”.

- Busca Binária:

- A busca binária é mais eficiente, mas ainda levará “log2(n)” passos, independentemente de onde “x” está na lista.

Cenário 4: X ∉ A

- Busca Linear:

- O pior caso ocorre quando “x” não está presente na lista. A busca linear precisa percorrer todos os “n” elementos.

- Busca Linear em Ordem:

- A busca linear em ordem pode sair antes, se encontrar um elemento maior que “x”. No pior caso, precisará percorrer “n” elementos.

- Busca Binária:

- A busca binária precisará dividir a lista até que o intervalo se esgote, resultando em “log2(n)” passos.

Com base nas análises acima, aqui está a tabela de contagem de tempo para os três algoritmos:

Situação	Busca Linear	Busca Linear em Ordem	Busca Binária
X ∈ A	n/2	n/2	Log2(n)
X = A[1]	1	1	O(1)
X = A[n]	n	n	Log2(n)
X ∉ A	n	n	Log2(n)

Resumo da Análise:

1. Busca Linear:

- Melhor caso: 1 passo.

- Pior caso: “n” passos.

- Caso médio: “n/2” passos.

2. Busca Linear em Ordem:

- Melhora marginal se o elemento não estiver na lista e a lista estiver ordenada, mas na prática o comportamento é semelhante à busca linear.

3. Busca Binária:

- Muito mais eficiente em listas ordenadas, com complexidade" log2(n)" para todos os casos, exceto o melhor caso trivial.