

Relatório T1: comparativo entre algoritmos de busca em um vetor.

1 – Análise das execuções:

Primeiramente, para analisarmos todos os 10 casos dados como entrada, foram implementados os três algoritmos de busca: binária, sequencial e bogo. Na tabela abaixo, podemos observar características como:

- Número de comparações: é o número de comparações feitas para testar se o número observado era, o número procurado. Esta coluna encontra-se dividida entre os resultados obtidos pelos três algoritmos acima citados.
- Índice encontrado: caso o algoritmo efetuasse com sucesso a busca, retornaria o índice de onde se encontrava o número procurado no vetor ou, caso o contrário, o valor -1. Assim como no número de comparações, esta coluna está dividida de acordo com o algoritmo utilizado.
- Ordenado (S/N): apresenta a letra S caso a entrada em um teste específico esteja ordenada, contém a letra N caso o contrário.

Caso/Algoritmo	Número de comparações:			Índice encontrado:			Ordenado (S/N):	N:
	Binária	Sequencial	Bogo	Binária	Sequencial	Bogo		
Teste 1	7	11	93	10	10	10	S	100
Teste 2	6	100	63	99	99	99	S	100
Teste 3	8	100	100	-1	-1	-1	S	100
Teste 4	1	501	157	500	500	500	S	1000
Teste 5	9	1	590	0	0	0	S	1000
Teste 6	17	14	10127	13	13	13	S	100000
Teste 7	17	90001	838	90000	90000	90000	S	100000
Teste 8	17	10000	81944	-1	9999	9999	N	100000
Teste 9	16	1001	82125	1000	1000	1000	S	100000
Teste 10	1	101	1	50000	100	89383	S	100000

(Tabela 1)

1.1 – Corretude:

Inicialmente, ao analisarmos os dados da Tabela 1, podemos afirmar claramente que em geral a busca binária obteve um desempenho melhor em relação aos outros dois algoritmos, visto que na maior parte dos testes foi quem menos efetuou instruções de comparação até encontrar o valor desejado. Porém, em um dos casos e mais especificamente no Teste 8, a mesma falhou ao efetuar a busca no vetor, e isso acontece pelo fato de a estrutura que armazena os dados não se encontrar ordenada, como podemos ver esse é o único caso onde ambos os fatores citados ocorrem. Logo, podemos garantir que sempre que o valor desejado se encontrar na estrutura, as buscas Bogo e Sequencial chegarão no resultado correto, diferentemente da Binária, que correrá o risco de entregar um resultado incorreto ou erro dependendo da implementação caso não satisfaça a condição de os dados estarem ordenados.

1.2 – Desempenho:

Agora, analisando os casos e seus resultados em geral, podemos deduzir alguns comportamentos dos seguintes cenários:

- **Cenário 1:** Em geral o algoritmo de busca binária se apresenta mais eficiente, isso porque na maioria dos casos apresentados, visto no Gráfico 1 que mostra o somatório de comparações de cada algoritmo, ele foi quem efetuou o menor número de comparações até



chegar ao resultado. E, além disso, também podemos observar que, de acordo com o aumento de N (número de valores na entrada), ele foi quem permaneceu com um número de comparações baixo, demonstrando que sua taxa de crescimento não é tão alta quanto a dos demais.

- **Cenário 2:** Casos onde o valor procurado se situava no início do vetor em posição que configurava pior caso para a busca binária, mais especificamente até a posição $\log(N) - 1$ visto que a busca binária continuamente olha somente para uma metade do vetor atual assumindo que estará neste caso o seu número de comparações será igual a $\log(N)$, a busca sequencial pode se sair melhor pelo fato de a binária se iniciar na metade do vetor e em seu pior caso testar este mesmo número de vezes mais um se o valor atual é o desejado. E no caso da bogo, sua probabilidade de que efetue um número de comparações menor do que o índice encontrado na sequencial é bem pequeno, sendo a sequencial, também, superior a esta.

- **Cenário 3:** Quando o valor procurado se situa no fim da estrutura a probabilidade de a busca sequencial ter o pior desempenho é maior, pois sempre terá que passar por todos os valores anteriores a ele antes de chegar no número desejado, ou testar todos caso o número não esteja no vetor. Logo, quanto mais para o fim o valor desejado estiver, maior o número de comparações feitas até se chegar a ele, e maior será a probabilidade desse algoritmo ter o pior desempenho entre os demais.

- **Cenário 4:** Em cenários onde o vetor encontra-se populado e grande parte dos números contidos nele tratam-se da repetição de um só valor, a busca Bogo pode se sair melhor do que a sequencial e até mesmo a Binária. Isso porque o fato de ela não seguir uma regra ao efetuar a busca faz com que a probabilidade de ela encontrar o valor desejado em cada comparação seja maior, visto que a mesma depende majoritariamente da quantidade de ocorrências do número desejado na estrutura. Ou seja, quanto maior o valor de N e do número de aparições do valor desejado no vetor, maior a probabilidade de a Bogo ser a mais eficiente.

2 – Escolha de algoritmo caso a caso:

Teste 1: Vetor ordenado e índice relativamente no início do vetor. Escolheria as buscas binária, sequencial e bogo, nesta ordem. Podemos dizer que as chances de a busca binária ser a mais eficiente são grandes, e o fato de o índice se encontrar mais no começo do vetor torna a busca sequencial, muito provavelmente, mais eficiente em relação a bogo. (Cenários 1 e 2)

Teste 2: Números em ordem e índice no fim do vetor. Optaria pelas buscas binária, bogo e sequencial, nesta ordem. Como primeira escolha teria a busca binária devido a sua constante eficiência para este tipo de caso e em seguida a bogo pelo fato de o número se encontrar no fim do vetor, assim tendo uma grande probabilidade de se sair melhor do que a sequencial. (Cenários 1 e 3)

Teste 3: Vetor em ordem e sem o valor desejado. Escolheria a busca binária como primeira opção e então as demais. Isso porque ela seria a única a não testar todos os valores para chegar no resultado. (Cenário 1)

Teste 4: Vetor ordenado e o valor desejado situado na metade do mesmo. Optaria pelas buscas binária, sequencial e bogo, nesta ordem. O seu posicionamento faz com que a busca binária faça apenas uma comparação, e, apesar de na execução apresentada pela tabela a busca bogo ter sido mais eficiente, escolheria a sequencial pois com a bogo eu sempre teria a chance de efetuar o dobro de comparações. (Cenários 1 e 3)

Teste 5: Valores em ordem e o número desejado encontra-se no índice 0. Escolheria pelas buscas sequencial, binária e bogo, nesta ordem. Neste caso a busca sequencial executa apenas uma comparação e a binária terá sempre o mesmo número de comparações, baixo, enquanto a bogo tem grandes chances de efetuar diversos testes até o resultado. (Cenários 2 e 1)

Teste 6: Vetor ordenado muito populado e o número desejado no início do mesmo. Escolheria as buscas sequencial, binária e bogo, nesta ordem. Isto porque o fato da estrutura ser muito grande ($N=100.000$) faz com que o índice encontrado, e consequentemente o número de comparações na busca sequencial, seja mais custoso na busca binária. (Cenários 2 e 1)

Teste 7: Entrada grande, estrutura em ordem e valor desejado situado no fim. Optaria pelas buscas binária, bogo e sequencial, nesta ordem. A busca binária, como nos casos em geral, muito provavelmente executaria o menor número de comparações. Já a bogo tem grandes possibilidades de se sair melhor em relação a sequencial, visto que o valor encontra-se bem ao fim do vetor. (Cenários 1 e 3)

Teste 8: Entrada grande, não ordenada e índice mais perto do início do vetor. Escolheria as buscas sequencial e bogo, nesta ordem. Neste caso a busca binária é desconsiderada pelo fato de haver uma chance enorme de obtermos o resultado errado, isso pois, diferentemente dos demais testes, os valores não estão ordenados. Quanto aos outros dois algoritmos, a busca sequencial tem uma probabilidade maior de ter um melhor desempenho visto que o valor desejado encontra-se no início do vetor. (Cenário 2)

Teste 9: Entrada grande, ordenada e valor desejado no início. Escolheria as buscas binária, sequencial e bogo, nesta ordem. Isto porque o fato do vetor ser grande faz com que a busca sequencial só tenha um melhor desempenho em relação a binária em índices menores, diretamente relacionado a taxa de crescimento no número de comparações citada anteriormente. E, como o número a ser encontrado encontra-se relativamente no início do vetor, a busca sequencial terá uma grande probabilidade de se sair melhor em relação a busca bogo. (Cenários 1 e 2)

Teste 10: O vetor tem tamanho elevado e está ordenado, o número desejado tem repetições que se iniciam em índice baixo e vão até o fim do vetor. Optaria pelas buscas binária, bogo e sequencial, nesta ordem. A busca binária, como em casos anteriores, tem grandes chances de ter o melhor desempenho, principalmente pelo fato da entrada ser grande. Neste caso específico, temos a repetição do valor a ser procurado um grande número de vezes, isso faz com que a bogo tenha uma probabilidade maior de ser mais eficiente em relação a sequencial, considerando que a mesma vai ter muitos índices onde encontrar o resultado e não seguir uma regra de busca. (Cenários 1 e 4)

3 – Conclusão:

O fato de o algoritmo de busca Bogo não seguir uma regra na hora de escolher qual índice será testado no passo seguinte faz com que as análises nesse relatório não possam ser exatas, porém há casos onde a partir das condições apresentadas pudemos avaliar, a partir da probabilidade, a possibilidade de um algoritmo ser superior a outro. Partindo deste raciocínio, notamos que as seguintes características têm forte influência no desempenho e corretude dos algoritmos:

- Localização do número procurado no vetor: há casos onde o índice se situa mais perto do início do vetor, favorecendo o desempenho de algoritmos como o de busca sequencial. Em contrapartida, quando o valor é encontrado no fim do vetor ou simplesmente não está lá, o algoritmo de busca sequencial se demonstra ineficiente.

- Ordenação dos dados de entrada: a utilização da busca Binária só deve ser considerada quando os valores dados, ou a estrutura que os armazena, estiverem ordenados. Isso porque seu funcionamento leva em consideração que essa condição ocorre, o que pode acarretar em erros ou até mesmo resultados incorretos caso a mesma não seja respeitada. Equanto isso, algoritmos de busca como a Bogo e Sequencial, garantem sua corretude mesmo em casos onde os dados não encontram-se ordenados, sempre encontrando o resultado **correto** a partir da entrada dada.

- Quantidade de valores na entrada: pudemos observar que a taxa de crescimento do número de comparações feitas pela busca binária não tem a mesma proporção da relacionada ao número de elementos no vetor. Diferentemente dela, em casos piores e médios casos, as buscas sequencial e bogo tem seus números de comparações aumentados de acordo com o crescimento da quantidade de valores do vetor.

- Repetição de valores no vetor: em casos onde há uma grande número de repetições do valor desejado no vetor de busca, as buscas binária e bogo têm grandes chances de serem mais eficientes, pois, diferentemente da sequencial, elas testam valores de diferentes áreas do vetor (início, meio e fim), aumentando as chances de que o valor seja encontrado.

Por fim, podemos dizer que, generalizadamente, nenhum dos algoritmos testados é superior ao outro. Isso porque a eficiência e corretude será sempre diretamente dependente das características dos casos onde os mesmos serão utilizados, sendo sempre recomendável efetuar uma análise prévia do cenário para que os resultados sejam encontrados corretamente e da forma mais eficiente o possível.