

Trabalho Final: Algoritmos de Busca

Algoritmos de busca são utilizados para verificar se um *dado* específico existe em uma sequência ou não. Por exemplo, em um vetor de números inteiros, com 1024 valores, denominado *Sequencia*, o algoritmo de busca deve verificar se, um valor *X* qualquer, ocorre nessa sequência.

Existem duas estratégias bastante conhecidas para implementação de algoritmos de busca: **sequencial e binária**. A primeira faz uma varredura por cada número do vetor, até encontrar o valor desejado, caso ele exista. A segunda estratégia, que exige um vetor previamente ordenado, a cada iteração, quebra o vetor ao meio: se o número a ser encontrado é inferior ao valor central do vetor, os números superiores a ele são descartados. Caso contrário, os números inferiores são descartados. A estratégia binária é aplicada diversas vezes na porção do vetor que não é descartada, até que se encontre o valor desejado, ou que se chegue à conclusão que o valor não estava presente no vetor.

Logo, a estratégia binária é mais eficiente que a estratégia sequencial, na maioria dos casos. Imagine que o número 8 deva ser procurado e ele não existe no vetor *Sequencia*. A estratégia sequencial precisará varrer os 1024 valores do vetor *Sequencia*, a fim de chegar a essa conclusão. Já a estratégia binária, no pior caso, terá que quebrar o vetor 10 vezes ($\log_2^{1024} = 10$).

Com base nos algoritmos de busca binária e sequencial:

- a) Desenvolva um programa com no mínimo três funções distintas: *busca sequencial*; *busca binária iterativa*; e *busca binária recursiva*.

As três funções devem receber por parâmetro o *Vetor Sequencia*, e o *valor* a ser procurado. As funções devem retornar 1, caso encontrem o valor desejado e 0, caso não encontrem.

- b) Devem ser criados quatro vetores para o armazenamento de valores inteiros, aleatórios, com tamanhos iguais a 10 mil, 50 mil, 100 mil e 200 mil números.
- c) Antes de chamar as duas funções com estratégia binária, os vetores precisam ser ordenados, utilizando o algoritmo com melhor desempenho visto em sala de aula.

- d) Demonstre, por meio de um gráfico, o desempenho (tempo de execução) dos três algoritmos de busca desenvolvidos. O gráfico deve mostrar o desempenho (em segundos), de cada algoritmo, ao procurar cinco valores distintos quaisquer, *presentes* nos vetores de 10, 50, 100 e 200 mil números.
- e) Demonstre, por meio de um gráfico, o desempenho (tempo de execução) dos três algoritmos de busca desenvolvidos. O gráfico deve mostrar o desempenho (em segundos), de cada algoritmo, ao procurar cinco valores distintos quaisquer, *não presentes* nos vetores de 10, 50, 100 e 200 mil números.
- f) Dado um vetor de números, de tamanho quinze, com os seguintes valores:

[1,20,32,41,58,62,71,88,97,100,112,115,120,180,199]

Mostre o funcionamento, detalhado, da pilha de execução (*call stack*) para a seguinte chamada à função de busca binária recursiva:

```
main() {  
    ...  
    retorno = busca_binaria_recursiva(vetor, 65);  
    ...  
}
```