

Tópicos Especiais

Algoritmos de Busca

Busca Linear

- Algoritmo de busca, também conhecido como Busca Sequencial.
- *Descrição*: Dado um vetor de elementos, checa cada um deles até que encontre o elemento procurado ou termine o vetor sem encontrar o elemento.
- *Aplicação*: Vetores ordenados e desordenados.
- *Eficiência*: Para um vetor com n elementos, faz:

- n comparações no pior caso (elemento procurado é o último do vetor).

Exemplo: Encontrar o número 3 no seguinte vetor com cinco números gasta cinco comparações.

5	2	1	4	3
---	---	---	---	---

- Uma comparação no melhor caso (elemento procurado é o primeiro do vetor). Exemplo:

Exemplo: Encontrar o número 5 no seguinte vetor com cinco números gasta uma comparação.

5	2	1	4	3
---	---	---	---	---

Algoritmo iterativo:

```
int buscaLinear(int *ptr_vet, int tamanho, int num)
{
    int passo = 0;
    int achou = 0;

    while (passo < tamanho && !achou) {
        if (*(ptr_vet + passo) == num)
            achou = 1;
        passo++;
    }
    return achou;
}
```

Algoritmo recursivo:

```
int buscaLinearRecursivo(int *ptr_vet, int tamanho, int num)
{
    if (tamanho == 1)
        if (num == *ptr_vet)
            return 1;
        else
            return 0;
    else
        if (num == *ptr_vet)
            return 1;
        else
            return buscaLinearRecursivo( ptr_vet + 1, tamanho - 1, num );
}
```

Busca Binária

- *Descrição:* Dado um vetor de elementos ordenados (ordem lexicográfica ou ordem crescente), começa examinando o elemento no centro do vetor. Se for igual ao elemento procurado, termina o algoritmo retornando verdadeiro. Senão, se o elemento no centro for maior, repete o processo de busca para a metade esquerda do vetor; se for menor, repete para a metade direita.
- *Aplicação:* Vetores ordenados e com possibilidade de acesso randômico aos elementos.
- *Eficiência:* Para um vetor com n elementos, faz:
 - $\lceil \log_2 n \rceil$ comparações no pior caso.

Exemplo: Encontrar o 5 no seguinte vetor com 5 números gasta 3 comparações ($\lceil \log_2 5 \rceil = 3$).

1	2	3	4	5
1	2	3	4	5
1	2	3	4	5

- Uma comparação no melhor caso (elemento procurado é o do centro do vetor).
Exemplo: Encontrar o 3 no seguinte vetor com 5 números gasta uma comparação.

1	2	3	4	5
---	---	---	---	---

Algoritmo iterativo:

```
int buscaBinaria(int *ptr_vet, int esq, int dir, int num)
{
    int meio, achou = 0;

    while (dir >= esq && !achou) {
        meio = (dir + esq) / 2;
        if (num > *(ptr_vet + meio))
            esq = meio + 1;
        else if (num < *(ptr_vet + meio))
            dir = meio - 1;
        else
            achou = 1;
    }
    return achou;
}
```

Algoritmo recursivo:

```
int buscaBinariaRecursivo(int *ptr_vet, int esq, int dir, int num)
{
    int meio;

    if (dir < esq)
        return 0;
    else {
        meio = (dir + esq) / 2;
        if (num > *(ptr_vet + meio))
            return buscaBinariaRecursivo(ptr_vet, meio + 1, dir, num);
        else if (num < *(ptr_vet + meio))
            return buscaBinariaRecursivo(ptr_vet, esq, meio - 1, num);
        else
            return 1;
    }
}
```