**1) Qual é objetivo da pesquisa em memória primária?**

Encontrar uma ou mais ocorrências de registros com chaves iguais à chave de pesquisa.

**2) Qual a diferença entre**[**Arquivos**](https://www.campusvirtual.ufsj.edu.br/portal/2018_2/mod/resource/view.php?id=8943)**e Tabelas?**

Tabela é associada a entidades de vida curta, criadas na memória interna durante a execução de um programa. E Arquivo é geralmente associado a entidades de vida mais longa, armazenadas em memória externa.

**3) O que influencia a escolha do método de Pesquisa?**

A quantidade dos dados envolvidos, e o Arquivo estar sujeito a inserções e retiradas frequentes.

**4) Quais são as operações mais comuns nos tipos abstratos de dados utilizados na Pesquisa em Memória Primária?**

1. Inicializar a estrutura de dados.

2. Pesquisar um ou mais registros com determinada chave.

3. Inserir um novo registro.

4. Retirar um registro específico.

5. Ordenar um arquivo para obter todos os registros em ordem d e acordo com a chave.

6. Ajuntar dois arquivos para formar um arquivo maior

**5) Quais são as operações do tipo abstrato de dados Dicionário?**

1. Inicializa

2. Pesquisa

3. Insere

4. Retira

**6) Descreva o algoritmo da pesquisa sequencial.**

function Pesquisa (x: TipoChave; var T: Tipotabela ) : TipoIndice;

var i : integer;

begin T.Item[0].Chave := x;

i := T.n + 1;

repeat i := i − 1; until T.Item[ i ] .Chave = x;

Pesquisa := i ;

end;

**7) Para que ser o registro sentinela na pesquisa sequencial?**

1. Garante que a pesquisa sempre termina: se o índice retornado por Pesquisa for zero, a pesquisa foi se m sucesso. 2. Não é necessário testar se i > 0, devido a isto: – o anel interno da função Pesquisa é extremamente simples: o índice i é decrementado e a chave de pesquisa é comparada com a chave que está no registro. – isto faz com que esta técnica seja conhecida como pesquisa sequencial rápida.

**8) Qual é custo do algoritmo de pesquisa sequencial no pior caso, melhor caso e caso médio?**

Melhor caso: C (n) = 1

Pior caso: C (n) = n

Médio caso: C (n) = (n + 1) / 2

**9) Em que situações é indicado o uso de pesquisa sequencial?**

O algoritmo de pesquisa sequencial é a melhor escolha para o problema de pesquisa em tabelas com até 25 registros.

**10) Descreva o algoritmo de pesquisa binária.**

function Binaria (x: TipoChave; var T: TipoTabela) : TipoIndice;

var i , Esq, Dir : TipoIndice;

begin

if T.n = 0

then Binaria := 0

else begin

Esq := 1; Dir := T.n;

repeat

i := (Esq + Dir ) div 2;

if x > T.Item[ i ] .Chave

then Esq := i+1

else Dir := i −1;

until (x = T.Item[ i ] .Chave) or (Esq > Dir );

if x = T.Item[ i ] .Chave then Binaria := i else Binaria := 0;

end;

end;

**11) Qual é a premissa básica para o funcionamento da pesquisa binária?**

Os registros estarem ordenados.

**12) Em que situações a pesquisa binária não é indicada?**

A pesquisa binária não deve ser usada em aplicações muito dinâmicas.

**13) Qual é custo do algoritmo de pesquisa binária no pior caso, melhor caso e caso médio?**

Pior caso: (n+1) / 2

Melhor caso: 1

Médio caso: (n+1) / 4

**14) Descreva o algoritmo de pesquisa em Árvore de Pesquisa.**

void pesquisa(struct registro \*r, struct arvore \*t) {

if(t == NULL) {

printf("Registro não esta presente na árvore\n");

}

else if(r->chave < t->reg.chave) {

pesquisa(r, t->esq);

}

else if(r->chave > t->reg.chave){

pesquisa(r, t->dir);

}

else {

\*x = t->reg;

}

}

**15) Quando é adequado utilizar Árvore de Pesquisa?**

É adequado quando existe a necessidade de considerar todos ou alguma combinação de:

1. Acesso direto e sequencial eficientes.

2. Facilidade de inserção e retirada de registros.

3. Boa taxa de utilização de memória.

4. Utilização de memória primária e secundária.

**16) Como é organizada a árvore? Qual é o critério para definir quem fica à direita ou à esquerda de**

**um nó?**

Para qualquer nó que contenha um registro, temos a relação invariante:

1. Todos os registros com chaves menores estão na sub árvore à esquerda.

2. Todos os registros com chaves maiores estão na sub árvore à direita.

**17) Qual é custo do algoritmo de pesquisa em Árvore de Pesquisa no pior caso, melhor caso e caso médio?**

Melhor caso: 1

Pior caso: Log n

Médio caso: (Log n)/2