Na operação de Remoção, não há a pesquisa. Parte do pressuposto de que a posição onde será removido já é conhecida.

Tabelas sem ordenação

Organizações:

- Compacta: deixar todos os registros válidos nas primeiras posições do vetor.

- Esparsa: não há a preocupação de onde está a localização, pode haver uma flag dizendo se aquele registro é válido ou não e na hora da inserção, ele insere em registros que estão marcados como falsos/inválidos.

Tabelas sem ordenação

**Compacta**: havendo *n* elementos, haverá dados da posição 0 a *n-1* da tabela.

criação

alocar cada elemento nas posições iniciais. complexidade *O(n)*

inserção

aloca o novo elemento na próxima posição livre e incrementa contador. *O(1)*

remoção

copia o ultimo registro sobre o que deve ser removido e decrementa o contador. *O(1)*

alteração

modifica o valor do campo. *O(1)*

pesquisa

pesquisa sequencial: consulta cada elemento pela chave procurada, parando quando achar ou chegar ao final. *O(n)*

manutenção

não existe.

*move to beginning*

troca o item pesq. com seu anterior, movendo p/ o início.

Deve-se escolher uma tabela de acordo com as operações que serão utilizadas, por exemplo, não há a necessidade de utilizar uma tabela extremamente eficiente em pesquisa sendo que esta operação quase nunca será utilizada. "Eu não preciso comprar uma bazuca pra matar uma mosca" - *Jander*

**Esparsa -** cada registro válido tem um marcador verdadeiro.

criação

alocação do registros nas posições inicias, marcando cada um como **verdadeiro** e as posições restantes como **falso**.

para i <- 0 a n-1

pega dado e coloca em i

fim-para

para i <- n até MAX-1

coloca F

fim-para

inserção

próxima posição vazia

remoção

marca com **falso**.

Tabelas Ordenadas

- chave de ordenação

- ordenação natural

criação

i) cria tabela vazia

insere cada registro mantendo a ordenação. *O(n^2)*

ii) coloca todos os registros na tabela

ordena  *O(n\*log n)*

inserção

Determina posição de inserção ( T(n) - pesquisa )

Deslocamento para abrir espaço ( *O(n)* )

remoção

Deslocamento dos registros com redução do contador. *O(n)*

alteração

se for alterada a chave de ordenação:

- dependendo da nova posição devem ser deslocados os registros para reacomodar o valor alterado. *O(n)*

manutenção: não existe.

pesquisa

pesquisa **sequencial**. *O(n)*

**sequencial ordenada**. *O(n)*

**binária.** *O(log n)***:**

inicio <- 0

fim <- n-1

enquanto fim-inicio > 0 faça

meio <- (inicio+fim)/2

se tab[meio].chave == chave então

fim <- inicio //força o fim

senão

se tab[meio].chave > chave então

fim <- meio-1

senão

inicio <- meio+1

fim-se

fim-se

fim-enquanto

se tab[meio].chave == chave então

retorna meio

**estimativa.** *O(log(log n))*