

1 -

FIFO (First-In, First-Out): Nessa política, a página que entrou primeiro na memória é a primeira a ser removida. É simples de implementar, porém pode sofrer do problema de "envelhecimento", em que páginas frequentemente referenciadas no início são removidas mesmo que sejam utilizadas com frequência.

LRU (Least Recently Used): Nessa política, a página que foi menos referenciada recentemente é removida. É considerada uma política eficiente, pois leva em conta o comportamento de acesso recente às páginas. No entanto, sua implementação pode exigir um custo adicional para acompanhar o histórico de acesso de cada página.

LFU (Least Frequently Used): Nessa política, a página que foi menos referenciada ao longo do tempo é removida. Ela leva em consideração a frequência de acesso de cada página e pode ser útil em cenários onde certas páginas são acessadas com menor frequência.

2 -

Organização dos dados: Os dados são organizados em uma sequência ordenada. Por exemplo, em uma lista, os elementos são dispostos em ordem crescente ou decrescente com base em algum critério.

Construção do índice: Um índice é criado para permitir um acesso rápido aos elementos da sequência. Esse índice pode ser uma estrutura de dados separada, como uma tabela hash ou uma árvore de busca, ou pode ser um vetor de índices que mapeia cada elemento para a sua posição na sequência.

Acesso através do índice: Com o índice construído, é possível acessar rapidamente um elemento específico da sequência. Basta usar o índice correspondente ao elemento desejado para localizá-lo diretamente.

Recebimento do valor a ser pesquisado: O usuário fornece o valor que deseja pesquisar na sequência.

Utilização do índice: O índice é consultado para determinar a posição inicial do acesso sequencial.

Comparação do valor: O valor pesquisado é comparado com o elemento na posição atual da sequência.

Verificação do resultado: Se o valor pesquisado for igual ao elemento na posição atual, a pesquisa é considerada concluída e o elemento é retornado. Caso contrário, verifica-se se o valor é menor ou maior do que o elemento atual.

Avanço na sequência: Com base na comparação, avança-se para o próximo elemento da sequência, repetindo os passos 3 e 4 até encontrar o elemento desejado ou percorrer toda a sequência.

3-

Ponteiro para o nó pai: Cada nó da árvore precisa armazenar um ponteiro para o seu nó pai, permitindo a navegação de cima para baixo na árvore.

Ponteiros para os filhos esquerdo e direito: Cada nó também precisa armazenar ponteiros para os seus filhos esquerdo e direito, permitindo a navegação de baixo para cima na árvore.

Organização por chave: Os nós são organizados com base em uma chave, que é um valor associado aos dados armazenados em cada nó. Os nós com chaves menores são posicionados à esquerda, enquanto os nós com chaves maiores são posicionados à direita.

Organização por ordem de inserção: Os nós são organizados na árvore de acordo com a ordem em que foram inseridos. Nesse caso, não há uma ordenação específica dos nós com base em suas chaves.

Organização por balanceamento: Além da chave, é aplicado um critério de balanceamento para garantir que a árvore permaneça balanceada, o que ajuda a manter a eficiência das operações. Exemplos de métodos de balanceamento incluem a Árvore AVL e a Árvore Rubro-Negra.

4-

Ordem da árvore (ordem mínima): A árvore B deve ter uma ordem mínima especificada, que determina o número mínimo de filhos que um nó pode ter e o número mínimo de chaves que um nó (exceto a raiz) pode armazenar.

Regra da página: Cada nó da árvore B é armazenado em uma página (bloco) de memória secundária. É importante respeitar a capacidade máxima de itens (chaves) que podem ser armazenados em uma página. A quantidade de itens por página deve ser definida de acordo com o espaço disponível na memória secundária.

Regra de preenchimento mínimo: Exceto a raiz, todos os nós não folha devem conter pelo menos metade do número máximo de chaves permitido. Isso garante que a árvore esteja bem balanceada e que as operações de busca e inserção sejam eficientes.

Regra de redistribuição: Durante as operações de inserção ou remoção que podem violar as regras anteriores, a redistribuição de chaves entre os nós é utilizada para manter a consistência da árvore B.

Regra da raiz: A raiz da árvore B pode ter no mínimo 1 chave e no máximo o número máximo de chaves permitido.

Regra de crescimento e divisão: Quando um nó está cheio e precisa ser dividido, ocorre o crescimento da árvore, onde um novo nó é criado e as chaves são redistribuídas entre os nós.