

Pesquisa Sobre Árvores Multiway

Árvore B

É uma árvore de busca balanceada multiway pensada para armazenamento em disco. Cada nó guarda várias chaves (entre um mínimo e máximo) e k chaves tem $k+1$ filhos, todas as folhas têm a mesma profundidade. Busca percorre níveis $O(\log n)$, inserção vai até a folha, se cheia, divide o nó (split) promovendo a chave do meio, remoção pode envolver redistribuição ou fusão de nós.

Árvore B+

Variação da B onde todas as chaves estão nas folhas, nós internos têm apenas chaves-guia. Folhas são encadeadas em lista para varreduras sequenciais eficientes. Inserção/remoção seguem a B, mas splits/merges ocorrem tipicamente nas folhas, ao dividir uma folha, uma chave-guia sobe ao pai. Lógica-chave: separar armazenamento de chaves e estrutura de navegação para otimizar consultas por intervalo e leituras sequenciais (muito usada em índices de bancos de dados).

Árvore B*

Melhora ocupação de páginas em relação à B. Invés de dividir imediatamente um nó cheio, tenta redistribuir chaves com irmãos via pai, só divide quando necessário (às vezes em split 3-way).

Árvore 2-3

É B-tree simples onde cada nó tem 1 ou 2 chaves (2 ou 3 filhos). Todas as folhas estão na mesma profundidade. Inserção: desce até a folha, se cheia (2 chaves), divide e promove a chave do meio, remoção usa fusão/redistribuição para manter limites.

Árvore 2-3-4

Permite nós com 1–3 chaves (2–4 filhos). Antes de descer durante a inserção, costuma-se dividir nós cheios (3 chaves) para evitar subir depois, split promove a chave do meio. Remoção combina fusões/redistribuições semelhantes.

Árvore Trie

Trie armazena strings por caracteres ao longo de caminhos da raiz; cada aresta representa um símbolo. Busca/inserção são $O(m)$ (m = comprimento da chave), independentemente do número de palavras. Tries são ideais para prefixos e autocompletar.

Problema principal: alto uso de memória (cada nó pode ter muitos filhos)

Árvore R

Indexa objetos espaciais agrupando objetos próximos em caixas delimitadoras mínimas (MBR). Cada entrada armazena um MBR, busca percorre ramos cujos MBRs podem conter a consulta. Inserção escolhe filho que minimize aumento de área, quando um nó excede capacidade, faz-se split com heurísticas para minimizar overlap. Lógica: agrupar espacialmente para responder consultas por área/interseção com poucos nós, eficiência depende muito da heurística de divisão.

Árvore R+

Variação que reduz sobreposição entre MBRs internas permitindo duplicação de objetos, um objeto pode aparecer em vários nós para manter regiões internas disjuntas. Lógica: trocar duplicação por menor overlap nos níveis superiores, reduzindo falsos positivos em buscas espaciais.

Arvore Patrícia (Variação da Trie)

PATRICIA é trie compactada que comprime caminhos sem ramificação em rótulos maiores (radix compression). Em vez de um nó por caractere, caminhos lineares são colapsados; buscas ainda seguem bits/caracteres, mas com menos nós. Inserção/remoção podem dividir/combina rótulos quando necessário.

Arvore Quad

Subdivide recursivamente um espaço 2D em quatro quadrantes (NE/NW/SE/SW). Para pontos/objetos, insere-se recursivamente no quadrante apropriado; se um quadrante

excede capacidade, subdivide-se novamente. Variantes (point, region, PR, MX) mudam critérios de divisão.