Árvores B

Livro: Estruturas de dados e seus algoritmos (pg: 160 – 170)
Autor: Jayme Luiz Szwarcfiter

Seja d um número natural. Uma árvore B de ordem d é uma árvore ordenada que é vazia, ou que satisfaz as seguintes condições:

1) **A raiz é uma folha ou tem no mínimo 2 filhos;**
2) **Cada nó diferente da raiz e das folhas possui no mínimo (d + 1) filhos;**
3) **Cada nó tem no máximo (2d + 1) filhos;**
4) **Todas as folhas estão no mesmo nível.**

Obs: Uma árvore 2-3 é uma árvore B de ordem 1.

Um nó de uma árvore B é chamado PÁGINA e armazena m chaves.

a) Uma página não folha tem (m + 1) filhos:

raiz - $1 \leq m \leq 2d$

outras - $d \leq m \leq 2d$

b) Em cada página P as chaves estão ordenadas $s_1 \leq s_2 \leq \ldots \leq s_M$ e contém (m + 1) ponteiros $p_0, p_1, \ldots, p_m$ para os filhos de P
(nas folhas estes ponteiros são NULL).

para qualquer chave y da página apontada por p_0, , $y < s_1$

Para qualquer chave y da página apontada por p_k, $1 \leq k \leq (m - 1)$, $s_k < y < s_{(k+1)}$

Para qualquer chave y da página apontada por p_m, $y > s_m$

Árvores B – armazenamento de grandes tabelas

(ler página 162 – Jayme)

Limites quanto ao número de páginas e elementos

Número mínimo de páginas: **ocorre quando a árvore possui uma página-raiz seguida do número mínimo de páginas para cada nível que se segue.**

Número máximo de páginas: **todas as páginas apontam para (2d + 1) páginas (filhos).**

Cotas extremas para a altura:

$$\log_{2d+1}(n+1) \leq h \leq 1 + \log_{d+1}(n+1/2)$$

Busca em árvores B : exemplo pag. 164

Inserção em árvores B

Passo 1: executar o procedimento de Busca
Passo 2: Se (chave já existe) então Inserção Inválida
Passo 3: senão inserir a chave na folha adequada

Problema: A folha já possuía 2d chaves, com a inserção obtemos 2d + 1 chaves (IMPOSSÍVEL POR DEFINIÇÃO)

Solução: Cisão de página

Seja P a página onde é feita uma inserção, resultando em (2d + 1) chaves armazenadas.

Disposição em P (apontada por pt)

p0, (s1,p1),(s2,p2),...,(sd,pd),(sd+1,pd+1),...,
...(s2d+1),p2d+1)

1. Em P permanecem d chaves
2. Criamos uma nova página Q apontada por pt1, a qual armazenará também d chaves
   pd+1,(sd+2,pd+2),...,(s2d+1,p2d+1)
3. Acrescentamos no nó W, pai de P, a entrada (sd+1,pt1), na ordem adequada.

Situação Possível: **W também pode ultrapassar o limite de chaves.**

Solução: **Propagar a Cisão até a raiz.**

Inserção em Árvores B

1. **Aplicar o procedimento de Busca, verificando a validade da inserção.**
2. **Se a inserção é válida, incluir a chave na folha F adequada.**
3. **Verificar se a página F necessita de cisão. Em caso positivo, propagar a cisão bottom-up enquanto necessário.**

Ver exemplo: página 166 de Szwarcfiter.
Ver applet na página: http://slady.cz/java/bt/

Remoção em Árvores B

Seja x a chave a ser removida. Temos 2 casos para considerar:

1. x se encontra em uma folha.
Solução: a entrada é simplesmente retirada;

2. x não se encontra em uma folha
Solução: x é substituída pelo sucessor (que também é folha)

(Logo, a análise da remoção pode se restringir ao caso em que esta operação é realizada em uma folha.)

Lembrando:

Os limites quanto ao número de chaves na página são:

raiz: $1 \leq m \leq 2d$

outras: $d \leq m \leq 2d$

Problemas na remoção

Quando a chave é retirada, podemos ter $(m < d)$.

Existem dois tratamentos

- **Concatenação e**
- **Redistribuição.**

Operação de Concatenação

Pré-condição:

Duas páginas podem ser concatenadas se são irmãs adjacentes e juntas possuem menos de 2d chaves.

Def.:

Duas páginas P e Q são chamadas *irmãs adjacentes* se têm o mesmo pai W e são apontadas por 2 ponteiros adjacentes em W.

Resultado da Concatenação:

1. **Agrupa a entrada das 2 páginas P e Q em uma apenas P;**
2. **Em W (pai) deixa de existir 1 entrada, exatamente aquela que existe entre os ponteiros adjacentes;**
3. **Esta chave passa a fazer parte da nova página concatenada P e seu ponteiro desaparece, visto que a página Q é devolvida;**
4. **Como uma chave foi retirada em W, esta página poderá ter menos do que d chaves e a concatenação é propagada (para cima).**

Acompanhe a operação....

- **Antes da concatenação**

Página W: $...(y_{j-1},pt),(y_j,pt1),(y_{j+1},pj+1),...$
Página P: $p0,(s1,p1),...,(sk,pk)$
Página Q: $po',(s1',p1'),...,(sm',pm')$ **onde** $k + m < 2d$

* **Depois da concatenação**

Página W: $...(y_{j-1},pt),(y_{j+1},pj+1),...$
Página P: $p0,(s1,p1),...,(sk,pk),(yj,po'),\ ,(s1',p1'),...,(sm',pm')$

ATENÇÃO: Concatenação é propagável (para cima), pois W tem menos chaves agora.

Operação de Redistribuição

Fato: **P e Q possuem em conjunto 2d ou mais chaves.**

Ação:

1. **Concatenação de P e Q (P fica grande demais)**
2. **Efetua-se uma cisão (use a página Q)**

ATENÇÃO:

Redistribuição não é propagável, pois W (pai) mantém o mesmo número de chaves.

Algoritmo Remoção (x)

Passo 1: **Aplicar o procedimento de Busca, verificando a existência da chave x na árvore. Seja P a página onde se encontra a chave.**

Passo 2: **Se P é uma folha, retirar a entrada correspondente a chave x. Caso contrário, buscar o sucessor. Seja z essa chave, e F a página onde z se encontra. Substitua a chave x por z. Fazer P = F.**

Passo 3: **Verificar se P contém d entradas. Caso contrário, executar a operação de concatenação ou redistribuição.**