Estruturas de Dados

Cristina Gomes Fernandes

B-árvores

Uma B-árvore T é uma árvore enraizada definida em função de um parâmetro t da seguinte maneira:

- todo nó q de T tem $n(q)$ chaves $k_1(q), \ldots, k_{n(q)}(q)$, com $n(q) \leq 2t - 1$ e $k_1(q) < \cdots < k_{n(q)}(q)$;
- para todo nó q, exceto a raiz, vale que $n(q) \geq t - 1$;
- as folhas de T estão todas no mesmo nível;
- cada nó interno q tem $n(q) + 1$ filhos $p_0(q), \ldots, p_{n(q)}(q)$;
- para todo nó q, as chaves da subárvore de raiz $p_i(q)$ estão todas entre $k_i(q)$ e $k_{i+1}(q)$, para $0 \leq i \leq n(q)$, considerando $k_0(q) = -\infty$ e $k_{n(q)+1}(q) = \infty$.

B-árvores

O parâmetro t é chamado de grau mínimo da árvore.

Quando $t = 2$, temos as chamadas árvores 2-3-4.

B-árvores

O parâmetro t é chamado de grau mínimo da árvore.

Quando $t = 2$, temos as chamadas árvores 2-3-4.

Lema: Uma B-árvore com $n > 0$ chaves e grau mínimo $t \geq 2$ tem altura $h \leq \log_t \frac{n+1}{2}$.

Ou seja, uma B-árvore com $t \geq 2$ tem altura logarítmica.

B-árvores

O parâmetro t é chamado de grau mínimo da árvore.

Quando $t = 2$, temos as chamadas árvores 2-3-4.

Lema: Uma B-árvore com $n > 0$ chaves e grau mínimo $t \geq 2$ tem altura $h \leq \log_t \frac{n+1}{2}$.

Ou seja, uma B-árvore com $t \geq 2$ tem altura logarítmica.

Na implementação, usamos um campo booleano $folha(q)$ para cada nó q da árvore, que vale VERDADE se e somente se q for uma folha.

B-árvores

O parâmetro t é chamado de grau mínimo da árvore.

Quando $t = 2$, temos as chamadas árvores 2-3-4.

Lema: Uma B-árvore com $n > 0$ chaves e grau mínimo $t \geq 2$ tem altura $h \leq \log_t \frac{n+1}{2}$.

Ou seja, uma B-árvore com $t \geq 2$ tem altura logarítmica.

Na implementação, usamos um campo booleano $folha(q)$ para cada nó q da árvore, que vale VERDADE se e somente se q for uma folha.

As rotinas DISKREAD(q) e DISKWRITE(q) são como explicado em aula.

Inicialização e busca em B-árvore

INICIALIZEBARV (r)

1 $r \leftarrow$ NOVACÉLULA$(0,$ VERDADE$)$

$\triangleright$ aloca r e faz $n(r) \leftarrow 0$ e $folha(r) \leftarrow$ VERDADE

2 DISKWRITE(r)

Inicialização e busca em B-árvore

INICIALIZEBARV (r)

$r \leftarrow$ NOVACÉLULA$(0,$ VERDADE$)$
  ▷ aloca r e faz $n(r) \leftarrow 0$ e $folha(r) \leftarrow$ VERDADE
DISKWRITE(r)

BUSQUEBARV (q, x) ▷ busca x na árvore de raiz q

$i \leftarrow 0$
enquanto $i < n(q)$ **e** $x \geq k_{i+1}(q)$ **faça**
  $i \leftarrow i + 1$
se $i \leq n(q)$ **e** $x = k_i(q)$
  então devolva (q, i)
se $folha(q)$
  então devolva $(\text{NIL}, 0)$
  senão DISKREAD$(p_i(q))$
      devolva BUSQUEBARV$(p_i(q), x)$

Inserção em B-árvore

INSIRABARV (r, x)

```
se n(r) = 2t − 1
   então z ← NOVACÉLULA(0, FALSO)      ▷ cria nova raiz
         p_0(z) ← r
         DIVIDAFILHO(z, 0)
         r ← z
INSIRANÃOCHEIO(r, x)
```

Inserção em B-árvore

DIVIDAFILHO (q, i) $\quad \triangleright$ q não está cheio mas $p_i(q)$ está cheio

$f \leftarrow p_i(q)$
$z \leftarrow$ NOVACÉLULA$(t-1, folha(f))$
para $j \leftarrow 1$ **até** $t-1$ **faça**
    $k_j(z) \leftarrow k_{j+t}(f)$
se NÃO $folha(f)$
    então para $j \leftarrow 0$ **até** $t-1$ **faça**
        $p_j(z) \leftarrow p_{j+t}(f)$
$n(f) \leftarrow t-1$
para $j \leftarrow n(q)$ **decrescendo até** $i+1$ **faça**
    $k_{j+1}(q) \leftarrow k_j(q)$
    $p_{j+1}(q) \leftarrow p_j(q)$
$k_{i+1}(q) \leftarrow k_t(f)$
$p_{i+1}(q) \leftarrow z$
$n(q) \leftarrow n(q) + 1$
DISKWRITE(q) DISKWRITE(f) DISKWRITE(z)

Inserção em B-árvore

INSIRANÃOCHEIO (q, x) $\triangleright$ q não está cheio

$i \leftarrow n(q)$
se $folha(q)$
    então enquanto $i \geq 1$ **e** $x < k_i(q)$ **faça**
            $k_{i+1}(q) \leftarrow k_i(q)$ $i \leftarrow i - 1$
        $k_{i+1}(q) \leftarrow x$
        $n(q) \leftarrow n(q) + 1$
        DISKWRITE(q)
    senão enquanto $i \geq 1$ **e** $x < k_i(q)$ **faça**
            $i \leftarrow i - 1$
        DISKREAD$(p_i(q))$
        se $n(p_i(q)) = 2t - 1$
            então DIVIDAFILHO(q, i)
                se $x > k_i(q)$
                    então $i \leftarrow i + 1$
        INSIRANÃOCHEIO$(p_i(q), x)$

Remoção em B-árvore

Exercício: Mostre como fica uma B-árvore com parâmetro $t = 2$ inicialmente vazia após a inserção de cada uma das chaves de 1 a 20.

Exercício: Escreva uma rotina que remove de uma B-árvore a chave mínima.

Exercício: Escreva a rotina de remoção de B-árvore, conforme a descrição feita em aula.

Exercício: Mostre como fica a B-árvore acima após a remoção da chave 5.

RemovaMin em B-árvore

REMOVAMINBARV (q) $\triangleright\ n(q) \geq t$

se $folha(q)$
  então $n(q) \leftarrow n(q) - 1$
    $x \leftarrow k_1(q)$
    para $i \leftarrow 1$ **até** $n(q)$ **faça**
      $k_i(q) \leftarrow k_{i+1}(q)$
  DISKWRITE(q)
    devolva x
  senão $f \leftarrow p_0(q)$
    se $n(f) = t - 1$
      então REARRANJABARV$(q, 0)$
    DISKREAD$(p_0(q))$
    devolva REMOVAMINBARV$(p_0(q))$