Algoritmos e Estruturas de Dados

árvores B

2010-2011

Carlos Lisboa Bento

Árvores B

conceitos

Múltiplos acessos a disco

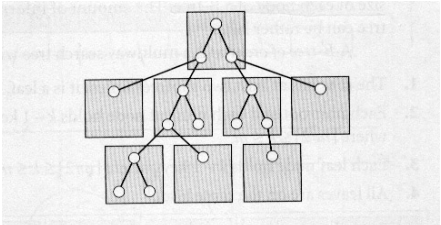
Tempos típicos de acesso

seek time: 40ms

latency (3000rpm): 10ms

data transfer rate: 10MB/s

As árvores binárias não são adequadas para manipulação de memória secundária

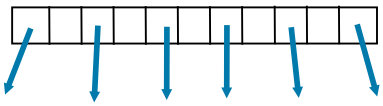


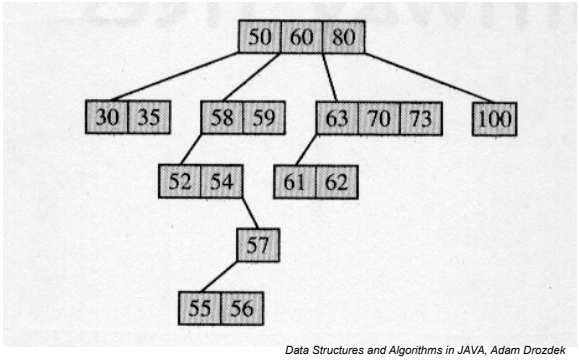
Data Structures and Algorithms in JAVA, Adam Drozdek

Árvores B
conceitos

ÁRVORES MULTICAMINHO

1. **Cada nó com m descendentes e m-1 chaves.**
2. **Chaves em cada nó ordenadas.**
3. **Chaves nos primeiros i descendentes menores que a i-ésima chave.**
4. **Chaves nos últimos m-i descendentes maiores que a i-ésima chave.**





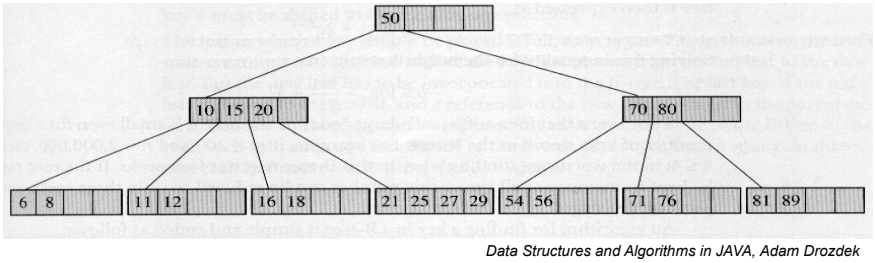
Data Structures and Algorithms in JAVA, Adam Drozdek

Árvores B
conceitos

ÁRVORES B

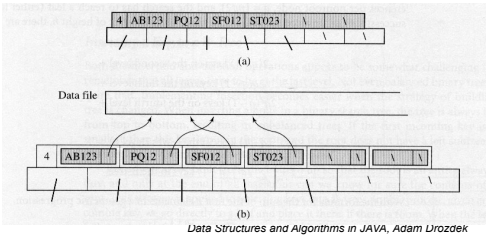
Uma árvore-B de ordem m é uma árvore m-ária com as seguintes propriedades:

1. **A raiz tem pelo menos 2 subárvores a menos que seja folha.**
2. **Os nós que não são raiz nem folha têm k-1 chaves e k referências para subárvores c/ $m/2 \leq k \leq m$.**
3. **Os nós folha têm k-1 chaves c/ $m/2 \leq k \leq m$.**
4. **Todas as folhas estão no mesmo nível.**



Data Structures and Algorithms in JAVA, Adam Drozdek

Árvores B
conceitos



Data Structures and Algorithms in JAVA, Adam Drozdek

```
class BTreeNode {
    int m = 4;
    boolean leaf = true;
    int keyTally = 1;
    int keys[] = new int[m-1];
    BTreeNode references[] = new BTreeNode[m];
    BTreeNode(int key) {
        keys[0] = key;
        for (int i = 0; i < m; i++)
            references[i] = null;
    }
}
```


Árvores B
conceitos

Procura

Pior caso – todos os nós com o n. mínimo de chaves,
1 na raiz e q = m/2 nos restantes nós.

Para uma árvore de altura h temos:

1 chave na raiz +

2 (q-1) chaves no nível 2 +

2 q (q-1) chaves no nível 3 +

2 q^2 (q-1) chaves no nível 4 +

...

2 q^{h-2} (q-1) chaves no nível h +

Temos então:

$$1+\left(\sum_{i=0}^{h-2} 2q^i\right)(q-1) \text{ chaves numa árvore B}$$

$$1+\left(\sum_{i=0}^{h-2} 2q^i\right)(q-1)=1+2(q-1)\left(\sum_{i=0}^{h-2} q^i\right)=1+2(q-1)\left(\frac{q^{h-1}-1}{q-1}\right)=-1+2q^{h-1}$$

Árvores B
conceitos

Procura

Pior caso – todos os nós com o n. mínimo de chaves,
1 na raiz e q = m/2 nos restantes nós.

Para uma árvore de altura h temos:

$$n \geq -1 + 2q^{h-1} \qquad h \leq \log_q \frac{n+1}{2} + 1$$

Ex.:

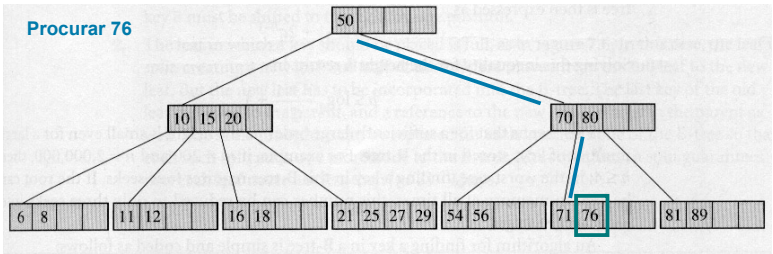
m=200 ➔ q=100

n=2 000 000

h ≤ 4

Árvores B
conceitos / implementação

Procura



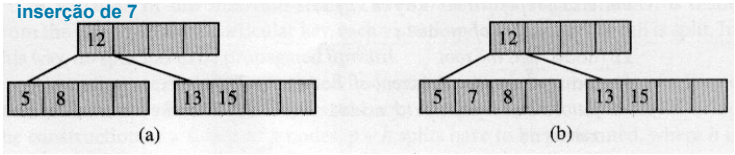

```
public BTreeNode BTreeSearch(int key) {
    return BTreeSearch(key,root);
}
protected BTreeNode BTreeSearch(int key, BTreeNode node) {
    if (node != null) {
        int i = 1;
        for ( ; i <= node.keyTally && node.keys[i-1] < key; i++);
        if (i > node.keyTally || node.keys[i-1] > key)
            return BTreeSearch(key,node.references[i-1]);

        else return node;
    }
    else return null;
}
```

Árvores B
conceitos

Inserção

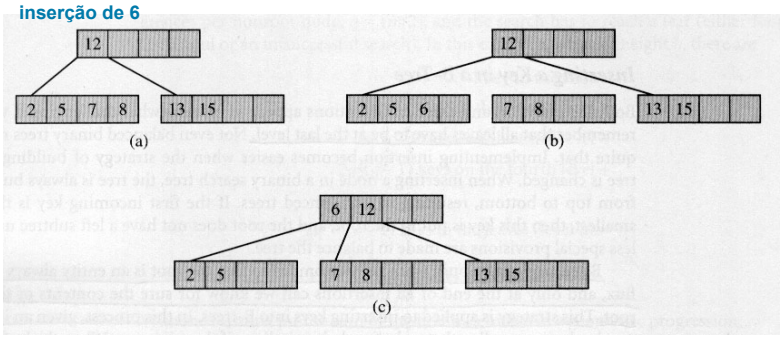
Caso 1 – inserção numa folha que ainda tem espaço livre

inserção de 7



Caso 2 – inserção que conduz a overflow numa folha

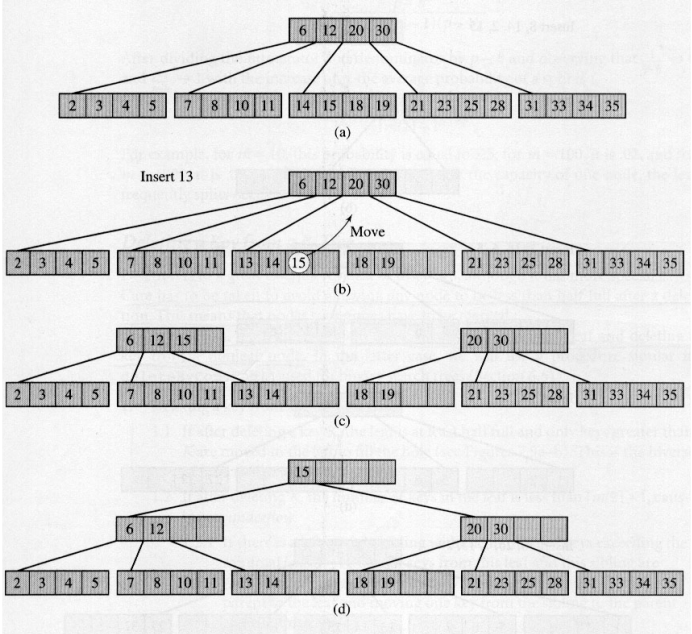
inserção de 6



Árvores B
conceitos

Caso 3 – inserção que conduz a overflow ao nível da raiz

inserção de 13



Árvores B
implementação

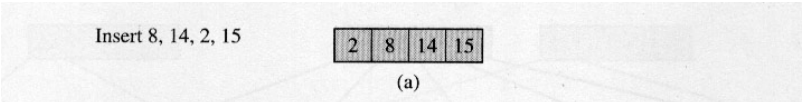
Inserção - Algoritmo

```
BTreeInsert (K)
  find a leaf node to insert K;
   while (true)
      find a proper position in array keys for K;
      if node is not full
         insert K and increment keyTally;
         return;
      else split node in node1 and node2;// node1 = node, node2 is new;
         distribute keys and references evenly between node1 and node2 and
         initialize properly their keyTally's;
         K = the last key of node1;
         if node was the root
            create a new root as parent of node1 and node2;
            put K and references to node1 and node2 in the root, and set its keyTally to 1;
            return;
         else node = its parent;// and now process the node's parent;
```


Árvores B
conceitos

Inserção

Exemplo árvore c/ m=5

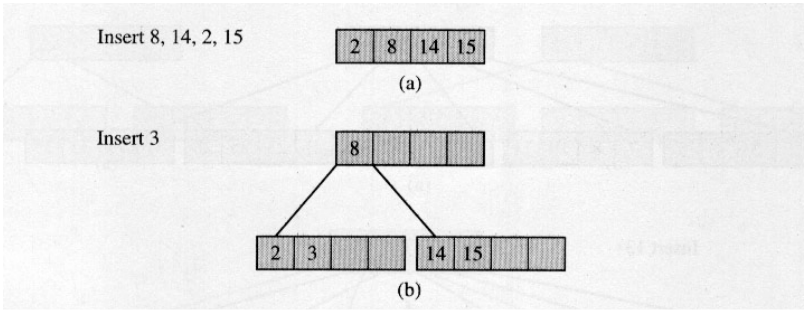


8 14 2 15 3 1 16 6 5 27
37 18 25 13 20 22 23 24

Árvores B
conceitos

Inserção
Exemplo árvore c/ m=5

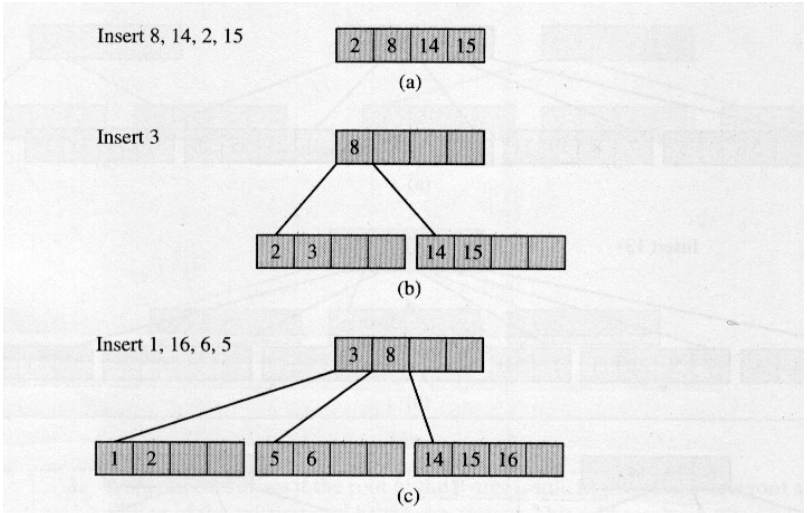
8 14 2 15 3 1 16 6 5 27
37 18 25 7 13 20 22 23 24



Árvores B
conceitos

Inserção
Exemplo árvore c/ m=5

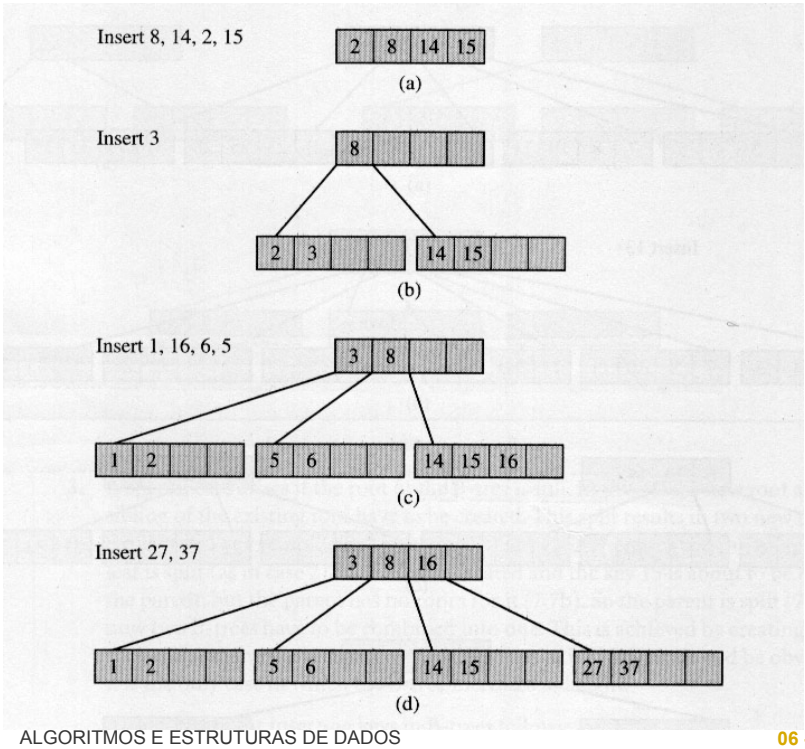
8 14 2 15 3 1 16 6 5 27
37 18 25 7 13 20 22 23 24



Árvores B
conceitos

Inserção
Exemplo árvore c/ m=5

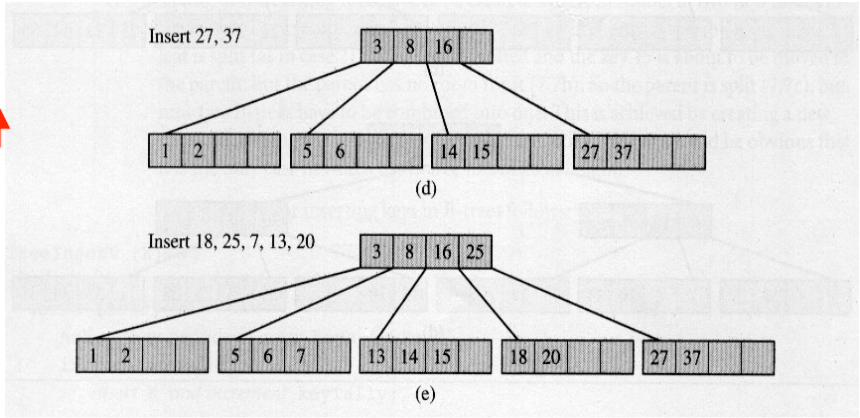
8 14 2 15 3 1 16 6 5 27
37 18 25 7 13 20 22 23 24



Árvores B
conceitos

Inserção
Exemplo árvore c/ m=5

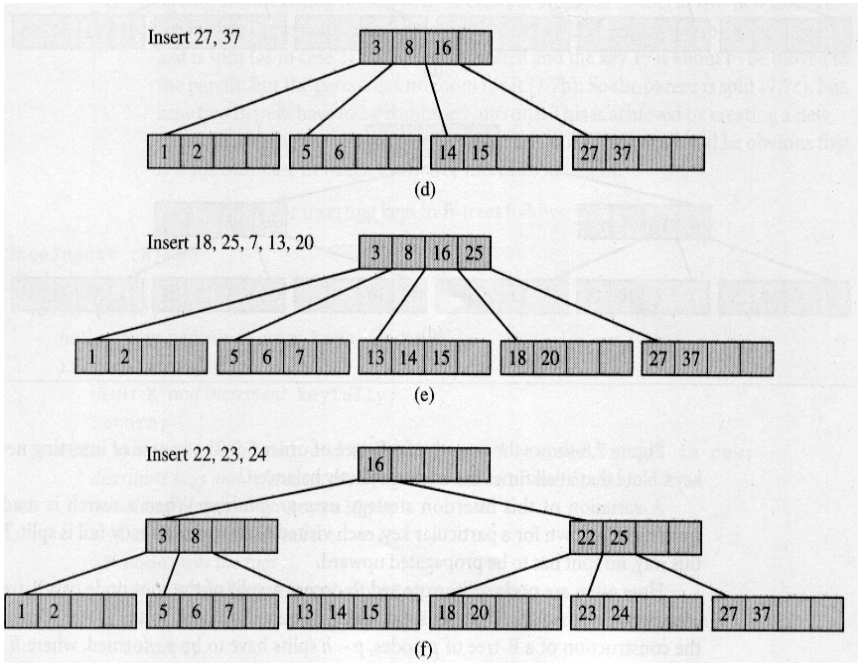
8 14 2 15 3 1 16 6 5 27 37
18 25 7 13 20 22 23 24



Árvores B
conceitos

Inserção
Exemplo árvore c/ m=5

8 14 2 15 3 1 16 6 5 27 37
18 25 7 13 20 22 23 24

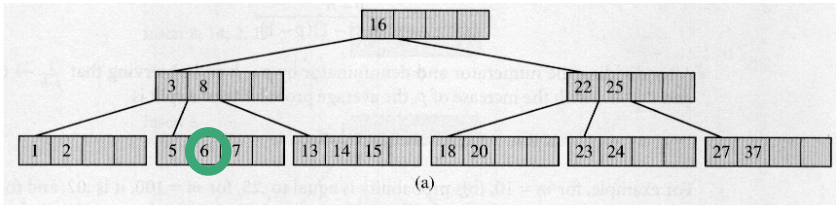


Árvores B
conceitos

Caso 1A – Numa folha em que esta não fica com menos de m/2 - 1 chaves

eliminação de 6

Eliminação
Exemplo árvore c/ m=5

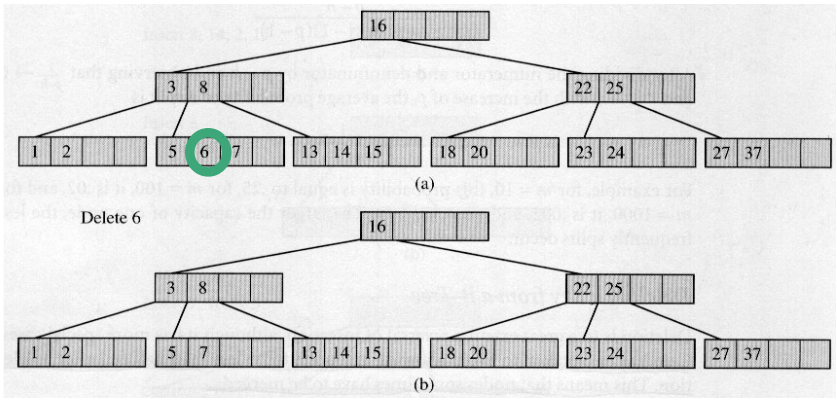


Árvores B
conceitos

Caso 1A – Numa folha em que esta não fica com menos de m/2 - 1 chaves

eliminação de 6

Eliminação
Exemplo árvore c/ m=5



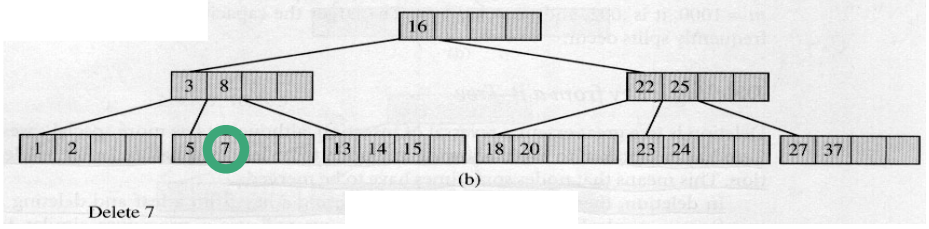
Árvores B
conceitos

Eliminação

eliminação de 7

Caso 1B – Numa folha em que esta fica com menos de m/2 - 1 chaves, mas pelo menos um dos seus irmãos não está no limite inferior do número de chaves (redistribuição)

Eliminação
Exemplo árvore c/ m=5



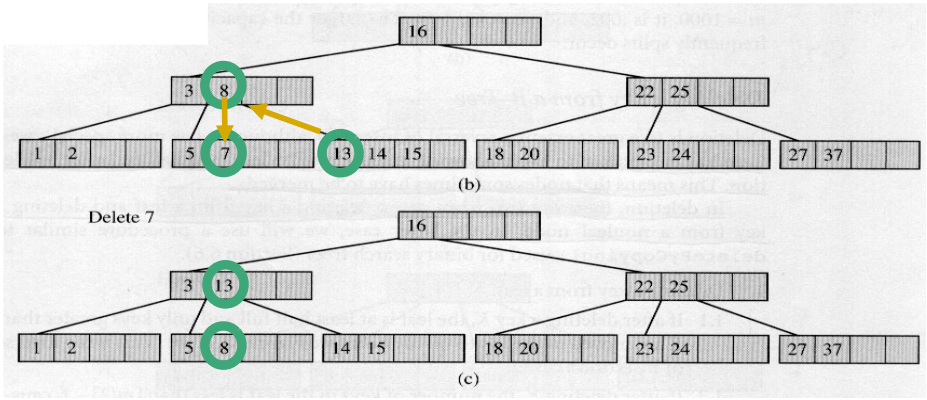
Árvores B
conceitos

Eliminação

Caso 1B – Numa folha em que esta fica com menos de m/2 - 1 chaves, mas pelo menos um dos seus irmãos não está no limite inferior do número de chaves (redistribuição)

eliminação de 7

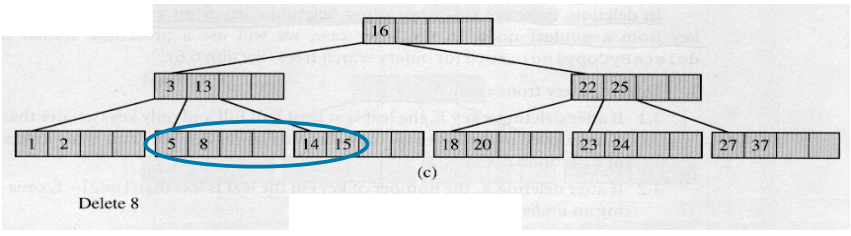
Eliminação
Exemplo árvore c/ m=5



Árvores B
conceitos

Caso 1C – Numa folha em que esta fica com menos de m/2 - 1 chaves e todos os seus irmãos estão no limite inferior do número de chaves

eliminação de 8



Eliminação
Exemplo árvore c/ m=5

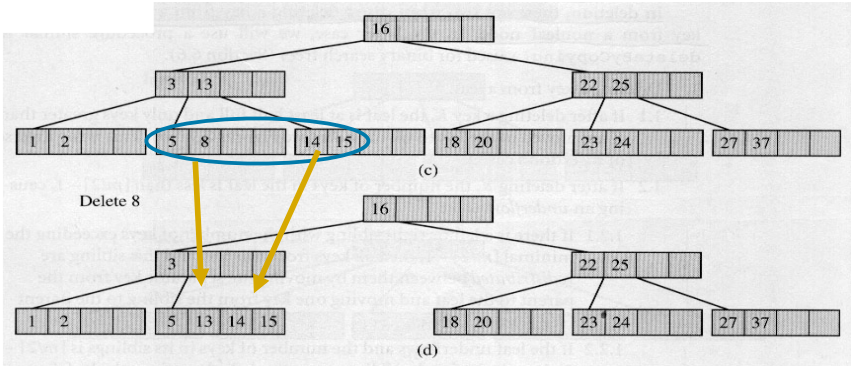
© DEI Carlos Lisboa Bento ALGORITMOS E ESTRUTURAS DE DADOS 06 -

Árvores B
conceitos

Caso 1C – Numa folha em que esta fica com menos de m/2 - 1 chaves e todos os seus irmãos estão no limite inferior do número de chaves

eliminação de 8

Eliminação
Exemplo árvore c/ m=5

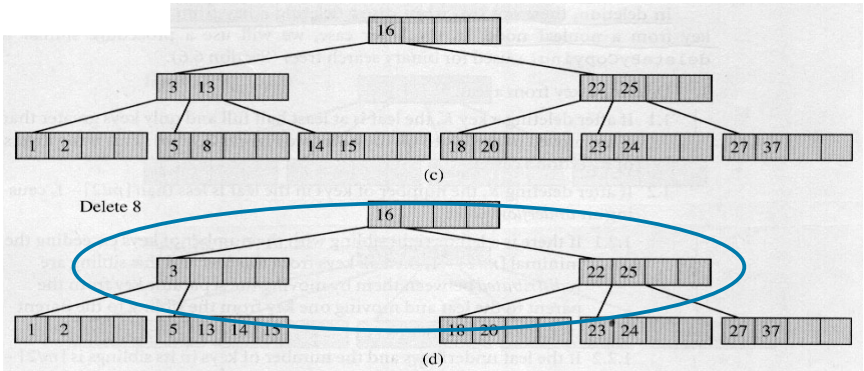


Árvores B
conceitos

Caso 1D – Numa folha ou não folha que resulte na fusão de nós em que o ascendente directo é uma raiz com uma única chave.

eliminação de 8

Eliminação
Exemplo árvore c/ m=5

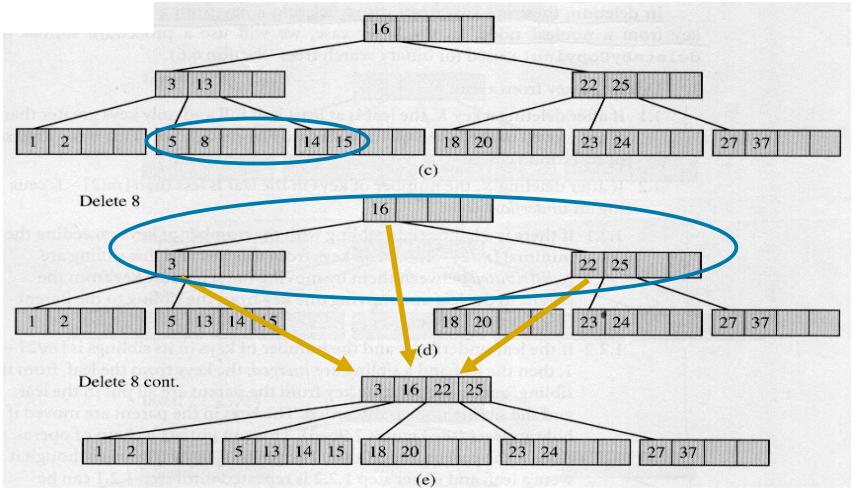


Árvores B
conceitos

Caso 1C – Numa folha em que esta fica com menos de m/2 - 1 chaves e todos os seus irmãos estão no limite inferior do número de chaves

eliminação de 8

Eliminação
Exemplo árvore c/ m=5



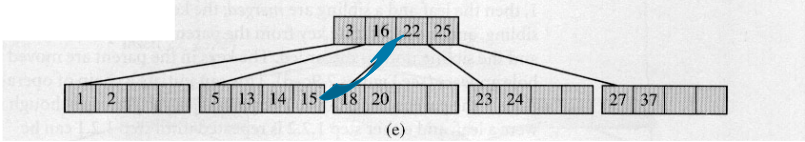
Árvores B
conceitos

Eliminação

Caso 2 – Num nó não folha. Vai ser reduzido ao problema de eliminar um nó de uma folha – caso contrário levaria a problemas de equilíbrio da árvore.

eliminação de 16

Eliminação
Exemplo árvore c/ m=5



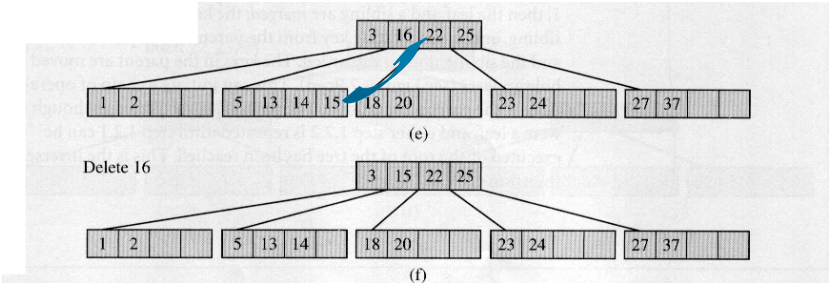
Árvores B
conceitos

Eliminação

Caso 2 – Num nó não folha. Vai ser reduzido ao problema de eliminar um nó de uma folha – caso contrário levaria a problemas de equilíbrio da árvore.

eliminação de 16

Eliminação
Exemplo árvore c/ m=5



Árvores B
conceitos

Eliminação - Algoritmo

```
BTreeDelete (K)
   node = BTreeSearch(K,root);
   if (node != null)
      if node is not a leaf                                   CASO 2
         find a leaf with the closest successor S of K;
         copy S over K in node;
         node = the leaf containing S;
         delete S from node;
      else delete K from node;
      while (true)
         if node does not underflow                           CASO 1A
            return;
         else if there is a sibling of node with enough keys  CASO 1B
            redistribute the keys between node and its sibling;
            return;
         else if node's parent is the root
            if the parent has only one key                    CASO 1D
               merge node, its sibling, and the parent to form a new root;
            else merge node and its sibling;                  CASO 1C
            return;
         else merge node and its sibling;
            node = its parent;
```

Árvores B
conceitos

Variantes das árvores B

- B*
- B+
- B+ Prefixas
- Bit
- R
- 2-4

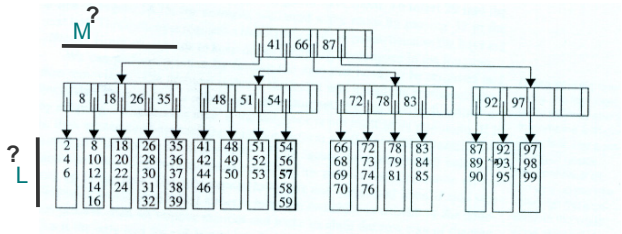
Árvores B
Demos na Web

http://slady.net/java/bt/

http://www.geocities.com/SiliconValley/Program/2864/File/btree.html

Árvores B

Exemplo cálculo de M



- Cada nó vai ocupar num bloco em disco
- Bloco = 8192 bytes
- Chave = 32 bytes
- M-1 chaves
- Cada nó (M-1)*32 + M Refs = 32*M - 32 + M Refs
- Ref = um número de referência para outro bloco em disco
- Referencia = 4 bytes
- Necessidades de memória para um nó não folha 36 * M - 32
- (8192+32) / 36 = 228,444 ➔ M=228

Árvores B

… end ;-)