

Árvores Digitais

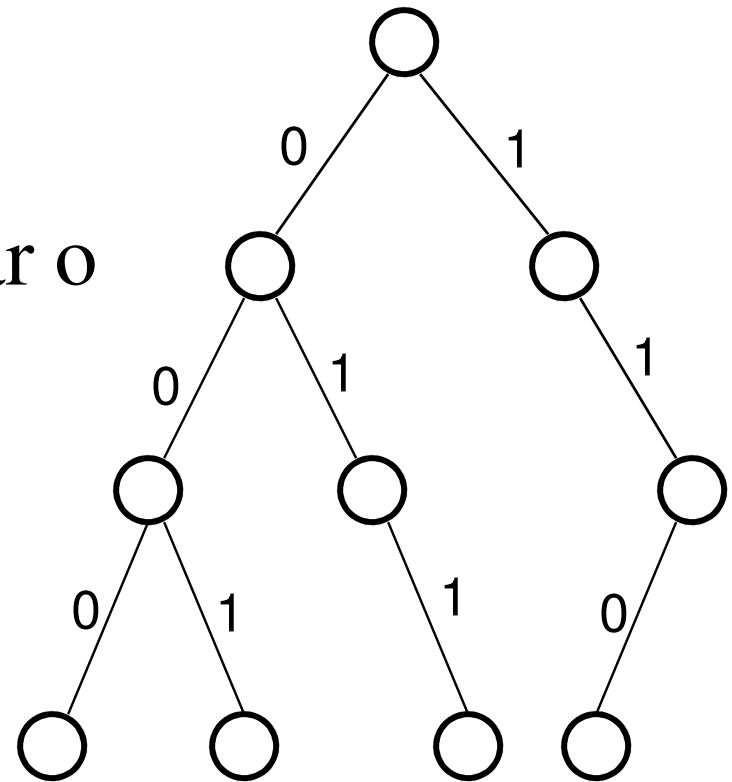
MC 202

Prof. Fernando Vanini

IC - UNICAMP

Árvores digitais

- Idéia: usar os bits do 'conteúdo' para indicar o 'caminho de busca':
 - 0 - esquerda
 - 1 - direita



Árvores digitais

- Vantagens
 - simplicidade
 - algoritmos são praticamente os mesmos de uma árvore binária de busca
- Desvantagens
 - altura
 - trechos comuns

Patrícia

- 'Practical Algorithm to Retrieve Information Coded In Alphanumeric)
- Idéia: cada nó tem um campo a mais indicando *o primeiro bit diferente*

0011001100

0011000011

↑
6

Se duas chaves tiverem um prefixo comum, a representação na árvore é feita a partir do primeiro bit diferente.

Patrícia

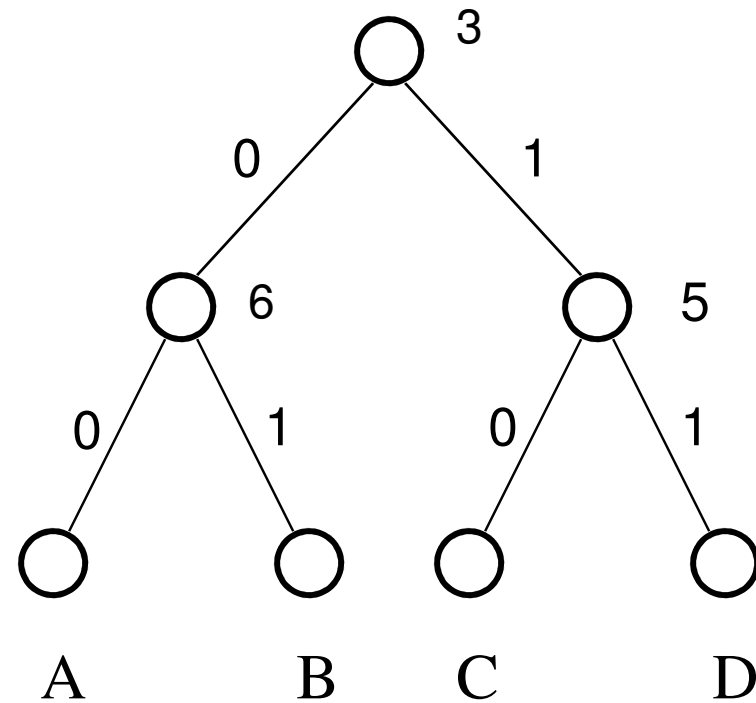
- Exemplo

10101101 A

10101111 B

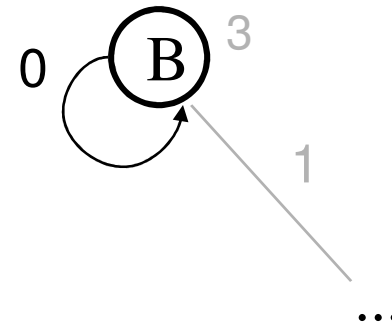
10110011 C

10110101 D



Patrícia

- Todo nó contém uma chave
- Assim que uma chave é inserida, o ramo da árvore correspondente ao bit diferente aponta para o nó recém criado.

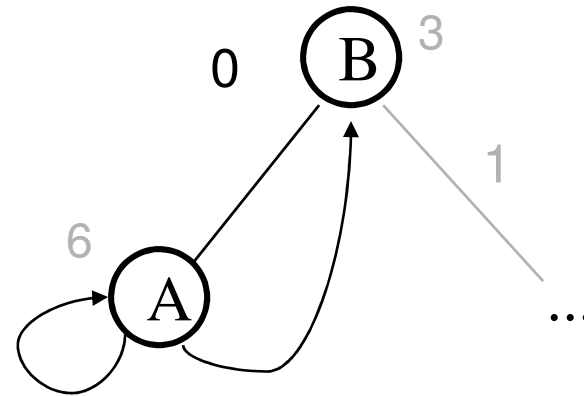


10101111 B

1011... ?

Patrícia

- Ao inserir uma nova chave, os apontadores são atualizados e as chaves antigas se mantêm na sua posição.



10101111 B

10101101 A

10110011 ?

Patrícia

```
typedef struct node *link;
typedef struct node{
    char *key;
    int b;
    link left, right;
} node;
```

```
int bit(char *str, int b) {
    if(str==NULL) return 0;
    char c = str[b/8];
    int bb = 1, i = 0;
    bb = 128 >> (b % 8);
    return (c & bb) != 0;
}
```


Patrícia

```
link patriciaSearch(char *v, link x) {  
    link p;  
    do{  
        p = x;  
        if(bit(v, x->b)) x = x->right;  
        else x = x->left;  
    }while (p->b < x->b);  
    return x;  
}
```

A função de acima só identifica um nó candidato. O string v deve ser comparado à chave associada ao nó.

Patrícia

```
link patriciaInsert(char *v, link x){
    link p; link t = patriciaSearch(v,x);
    if(strcmp(v,t->key) != 0){
        int i = 0;
        while(bit(v,i) == bit(t->key,i)) i++;
        do{
            p = x;
            if(bit(v,x->b)) x = x->right; else x = x->left;
        }while((x->b < i) && (p->b < x->b));
        t = malloc(sizeof(node));
        t->key = v; t->b = i;
        if(bit(v,i)) { t->right = t; t->left = x; }
        else { t->left = t; t->right = x; }
        if(bit(v,p->b)) p->right = t; else p->left = t;
    }
    return t;
}
```

Patrícia

- Características
 - Para uma chave com k bits, numa busca:
 - são feitas no máximo k comparações de bits
 - apenas uma comparação de de chaves é necessária.
 - A altura média de uma árvore com N nós é $\log_2 N$

Referências na web

- <http://www.csse.monash.edu.au/~lloyd/tildeAlgDS/Tree/PATRICIA/>
- http://en.wikipedia.org/wiki/Radix_tree
- <http://www.freepatentsonline.com/6012061.html>