



Estruturas de Busca em Texto

ESTRUTURA DE DADOS II

Busca em Texto

Busca em texto, ou pesquisa digital, ou 'casamento de padrões', ou casamento de cadeias, consiste na pesquisa de uma subsequência de símbolos numa sequência destes (símbolos).

Busca em Texto

Pode ter como objetivo encontrar todas as ocorrências, ou somente a primeira, de uma dada subsequência numa dada sequência; a exemplo, de um palavra em um texto.



Onde/Quando
se aplica
busca em
texto?

Busca em Texto

A busca em texto é útil para gerenciamento de editores de textos, dicionários, recuperação de dados, manipulação de símbolos.



Como efetuar
busca em
texto?

Busca em Texto

Na busca digital, diferente do que foi visto até então, a chave não é tratada como um elemento único, indivisível. Assume-se que cada chave é constituída de um conjunto de caracteres ou dígitos.

Busca em Texto

Assim, como a chave não é tratada como elemento único, ao invés de se comparar a chave procurada com as chaves do conjunto armazenado, a comparação é efetuada caractere a caractere, dentre os caracteres que compõem as chaves.

Busca em Texto

Para efetuar **busca em texto**, é possível explorar:

Algoritmos

KMP (Knuth, Moris e Pratt)
BM (Boyer-Moore)
Rabin-Karp

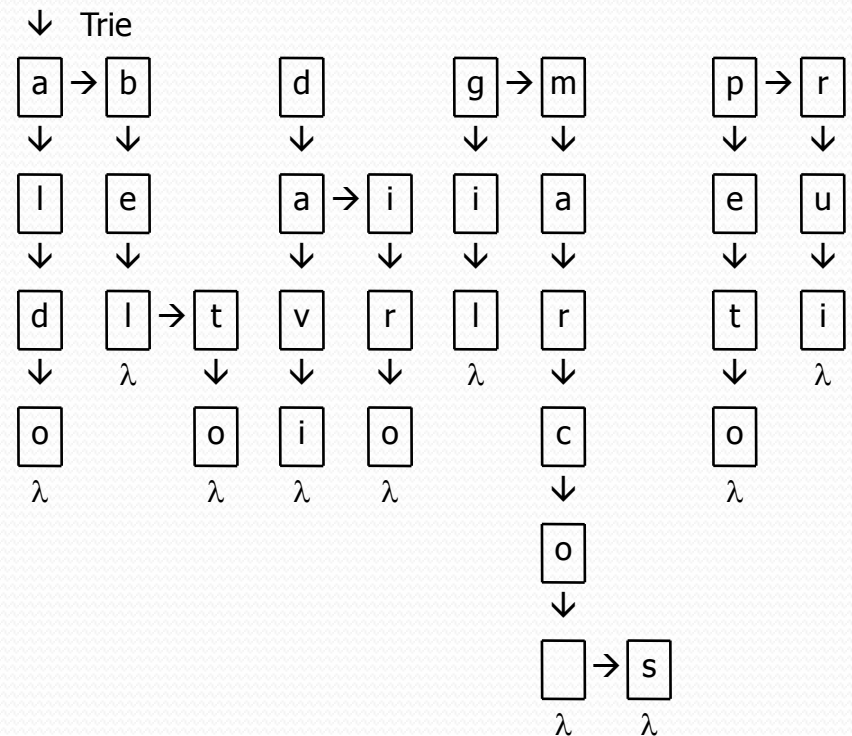
Estruturas de Dados

Trie
Árvore Digital

TRIE

Termo proveniente da palavra inglesa **retrieval** (recuperação), e corresponde a uma estrutura usada para recuperação de dados – de subsequência, em sequência.

	1	2	3	4	...9	10
a	aldo	davi			marco	
b	3					
d	2					
e			4			
g	gil					
i		10				
l				bel		
m	5					
n						dino
p	peto					
q						
r	rui					diro
s					marcos	
t				beto		



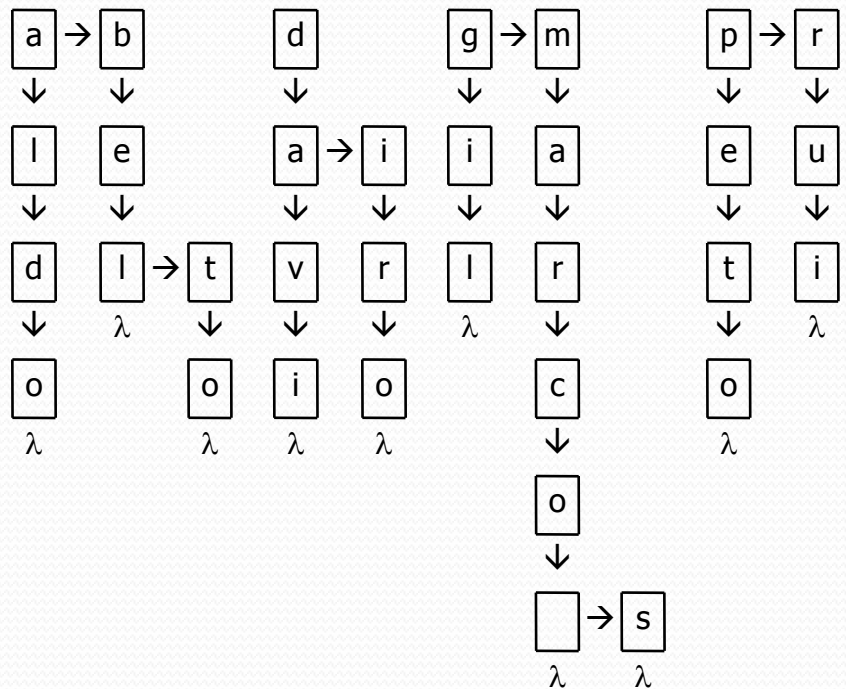
TRIE



Qual a pronúncia correta?

	1	2	3	4	...9	10
a	aldo	davi			marco	
b	3					
d	2					
e			4			
g	gil					
i		10				
l				bel		
m	5					
n						dino
p	peto					
q						
r	rui					diro
s					marcos	
t				beto		

↓ Trie

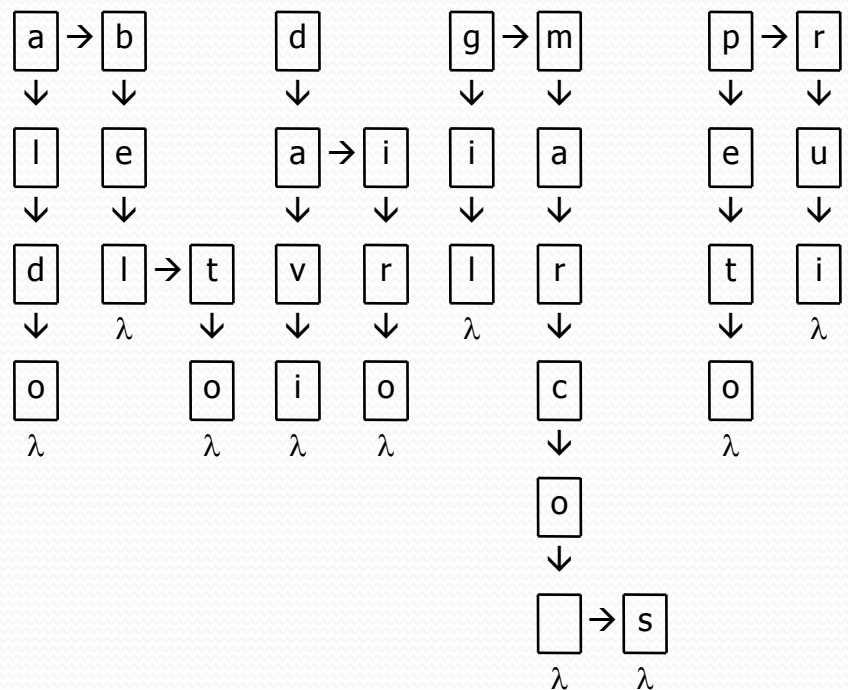


TRIE

São estruturas em que as chaves são tratadas caractere a caractere. E podem ser implementadas por meio de estruturas sequenciais ou dinâmicas.

	1	2	3	4	...9	10
a	aldo	davi			marco	
b	3					
d	2					
e			4			
g	gil					
i		10				
l				bel		
m	5					
n						dino
p	peto					
q						
r	rui					diro
s					marcos	
t				beto		

↓ Trie



Construindo uma **trie estática** para armazenamento das palavras:

1. davi
2. aldo
3. beto
4. diro
5. peto
6. rui
7. bel
8. gil
9. marco
10. marcos
11. dino

Reserva-se:

	1	2	3	4	5	6
a						
b						
d						
e						
g						
i						
l						
m						
n						
p						
q						
r						
s						
t						

Para economia de espaço algumas letras foram suprimidas.

Trie Estática

1

2

3

4

5

6



Onde inserir
davi?

	1	2	3	4	5	6
a						
b						
d						
e						
g						
i						
l						
m						
n						
p						
q						
r						
s						
t						

Construindo uma **trie estática** para armazenamento das palavras:

1. davi
2. aldo
3. beto
4. diro
5. peto
6. rui
7. bel
8. gil
9. marco
10. marcos
11. dino

Inserir **aldo**.

	1	2	3	4	5	6
a						
b						
d	davi					
e						
g						
i						
l						
m						
n						
p						
q						
r						
s						
t						

Construindo uma **trie estática** para armazenamento das palavras:

1. davi
2. aldo
3. beto
4. diro
5. peto
6. rui
7. bel
8. gil
9. marco
10. marcos
11. dino

Inserir **beto**.

	1	2	3	4	5	6
a	aldo					
b						
d	davi					
e						
g						
i						
l						
m						
n						
p						
q						
r						
s						
t						

Construindo uma **trie estática** para armazenamento das palavras:

1. davi
2. aldo
3. beto
4. diro
5. peto
6. rui
7. bel
8. gil
9. marco
10. marcos
11. dino

Inserir **diro**.

	1	2	3	4	5	6
a	aldo					
b	beto					
d	davi					
e						
g						
i						
l						
m						
n						
p						
q						
r						
s						
t						

Construindo uma **trie estática** para armazenamento das palavras:

1. davi
2. aldo
3. beto
4. diro
5. peto
6. rui
7. bel
8. gil
9. marco
10. marcos
11. dino

Inserir **peto**.

	1	2	3	4	5	6
a	aldo	davi				
b	beto					
d	2					
e						
g						
i		diro				
l						
m						
n						
p						
q						
r						
s						
t						

Construindo uma **trie estática** para armazenamento das palavras:

1. davi
2. aldo
3. beto
4. diro
5. peto
6. rui
7. bel
8. gil
9. marco
10. marcos
11. dino

Inserir **rui**.

	1	2	3	4	5	6
a	aldo	davi				
b	beto					
d	2					
e						
g						
i		diro				
l						
m						
n						
p	peto					
q						
r						
s						
t						

Construindo uma **trie estática** para armazenamento das palavras:

1. davi
2. aldo
3. beto
4. diro
5. peto
6. rui
7. bel
8. gil
9. marco
10. marcos
11. dino

Inserir **bel.**

	1	2	3	4	5	6
a	aldo	davi				
b	beto					
d	2					
e						
g						
i		diro				
l						
m						
n						
p	peto					
q						
r	rui					
s						
t						

Construindo uma **trie estática** para armazenamento das palavras:

1. davi
2. aldo
3. beto
4. diro
5. peto
6. rui
7. bel
8. gil
9. marco
10. marcos
11. dino

Inserir **gil**.

	1	2	3	4	5	6
a	aldo	davi				
b	3					
d	2					
e			4			
g						
i		diro				
l				bel		
m						
n						
p	peto					
q						
r	rui					
s						
t				beto		

Construindo uma **trie estática** para armazenamento das palavras:

1. davi
2. aldo
3. beto
4. diro
5. peto
6. rui
7. bel
8. gil
9. marco
10. marcos
11. dino

Inserir **marco**.

	1	2	3	4	5	6
a	aldo	davi				
b	3					
d	2					
e			4			
g	gil					
i		diro				
l				bel		
m						
n						
p	peto					
q						
r	rui					
s						
t				beto		

Construindo uma **trie estática** para armazenamento das palavras:

1. davi
2. aldo
3. beto
4. diro
5. peto
6. rui
7. bel
8. gil
9. marco
10. marcos
11. dino

Inserir **marcos** e **dino**.

	1	2	3	4	5	6
a	aldo	davi				
b	3					
d	2					
e			4			
g	gil					
i		diro				
l				bel		
m	marco					
n						
p	peto					
q						
r	rui					
s						
t				beto		

Construindo uma **trie estática** para armazenamento das palavras:

1. davi
2. aldo
3. beto
4. diro
5. peto
6. rui
7. bel
8. gil
9. marco
10. marcos
11. dino

	1	2	3	4	...9	10
					marco	
a	aldo	davi				
b	3					
d	2					
e			4			
g	gil					
i		10				
l				bel		
m	5					
n						dino
p	peto					
q						
r	rui					diro
s					marcos	
t				beto		

Trie Estática



Como efetuar
consulta?

	1	2	3	4	...9	10
					marco	
a	aldo	davi				
b	3					
d	2					
e			4			
g	gil					
i		10				
l				bel		
m	5					
n						dino
p	peto					
q						
r	rui					diro
s					marcos	
t				beto		

Trie Estática



Como, através
desta, por
exemplo,
manter um
dicionário?

	1	2	3	4	...9	10
					marco	
a	aldo	davi				
b	3					
d	2					
e			4			
g	gil					
i		6				
l				bel		
m	5					
n						dino
p	peto					
q						
r	rui					diro
s					marcos	
t				beto		

Trie Estática



Esta estrutura
facilita a
implementação
do “**auto-
completar**”?

	1	2	3	4	...9	10
					marco	
a	aldo	davi				
b	3					
d	2					
e			4			
g	gil					
i		6				
l				bel		
m	5					
n						dino
p	peto					
q						
r	rui					diro
s					marcos	
t				beto		

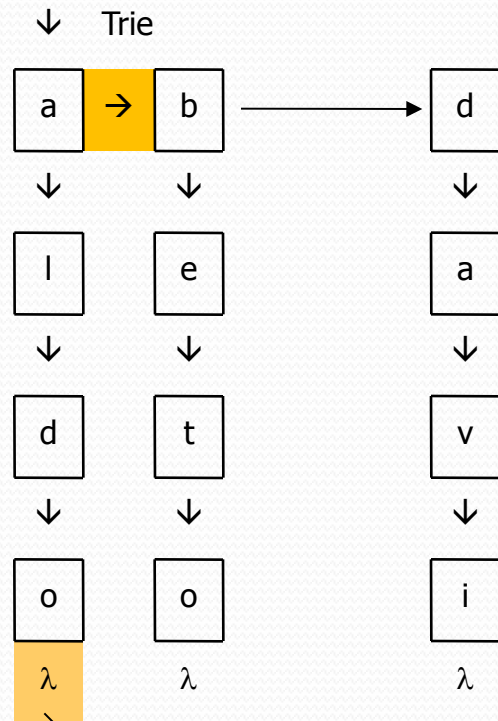
Trie Estática



Como
dimensionar
esta estrutura?

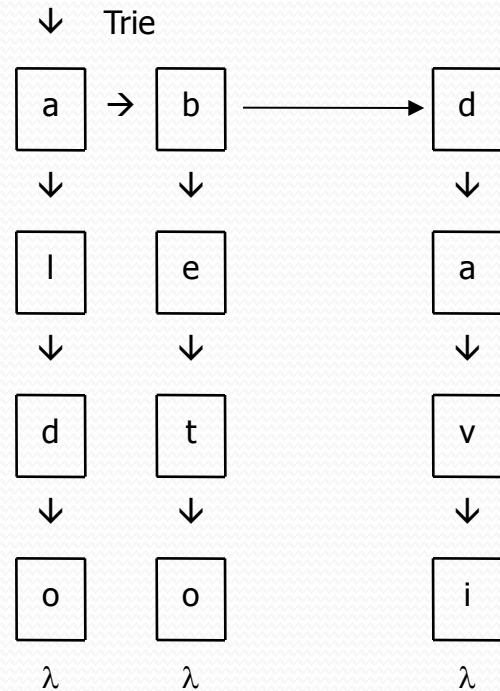
	1	2	3	4	...9	10
					marco	
a	aldo	davi				
b	3					
d	2					
e			4			
g	gil					
i		6				
l				bel		
m	5					
n						dino
p	peto					
q						
r	rui					diro
s					marcos	
t				beto		

Construindo
uma **trie**
dinâmica
(cada letra é
mantida
numa
variável
dinâmica)
para
armazenar:
davi, aldo,
beto, diro,
peto, rui, bel,
gil , marco,
marcos, dino.



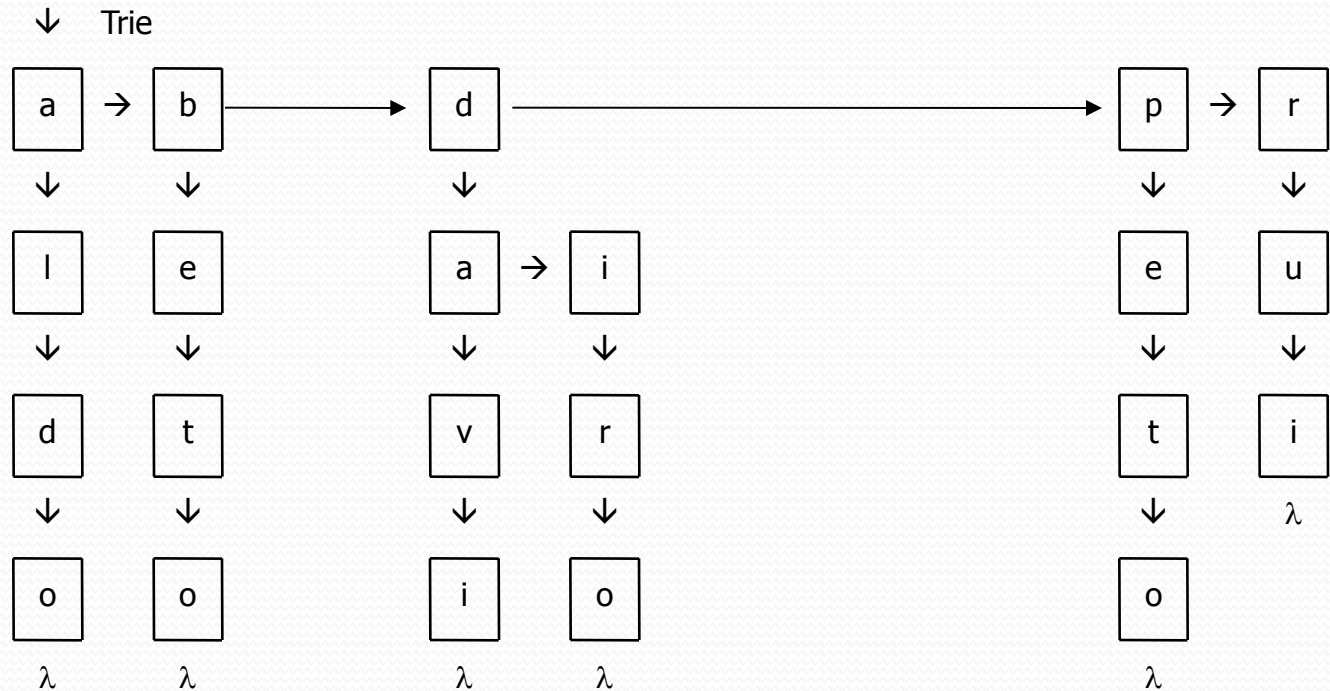
Há aterramento também dos ponteiros “laterais”.

Construindo uma **trie dinâmica** (cada letra é mantida numa variável dinâmica) para armazenar: davi, aldo, beto, diro, peto, rui, bel, gil, marco, marcos, dino.

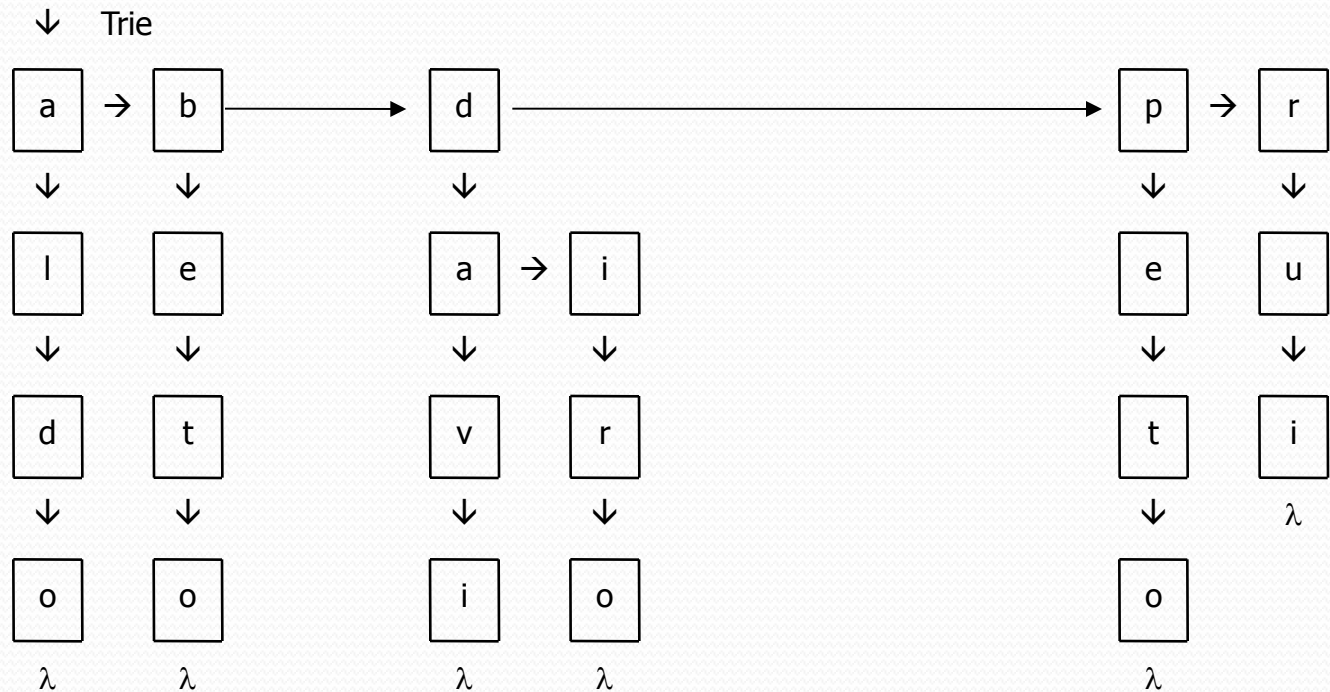


Inserir **diro**,
peto e **rui**?

Construindo
uma **trie**
dinâmica
(cada letra é
mantida
numa variável
dinâmica)
para
armazenar:
davi, aldo,
beto, diro,
peto, rui, bel,
gil , marco,
marcos, dino.

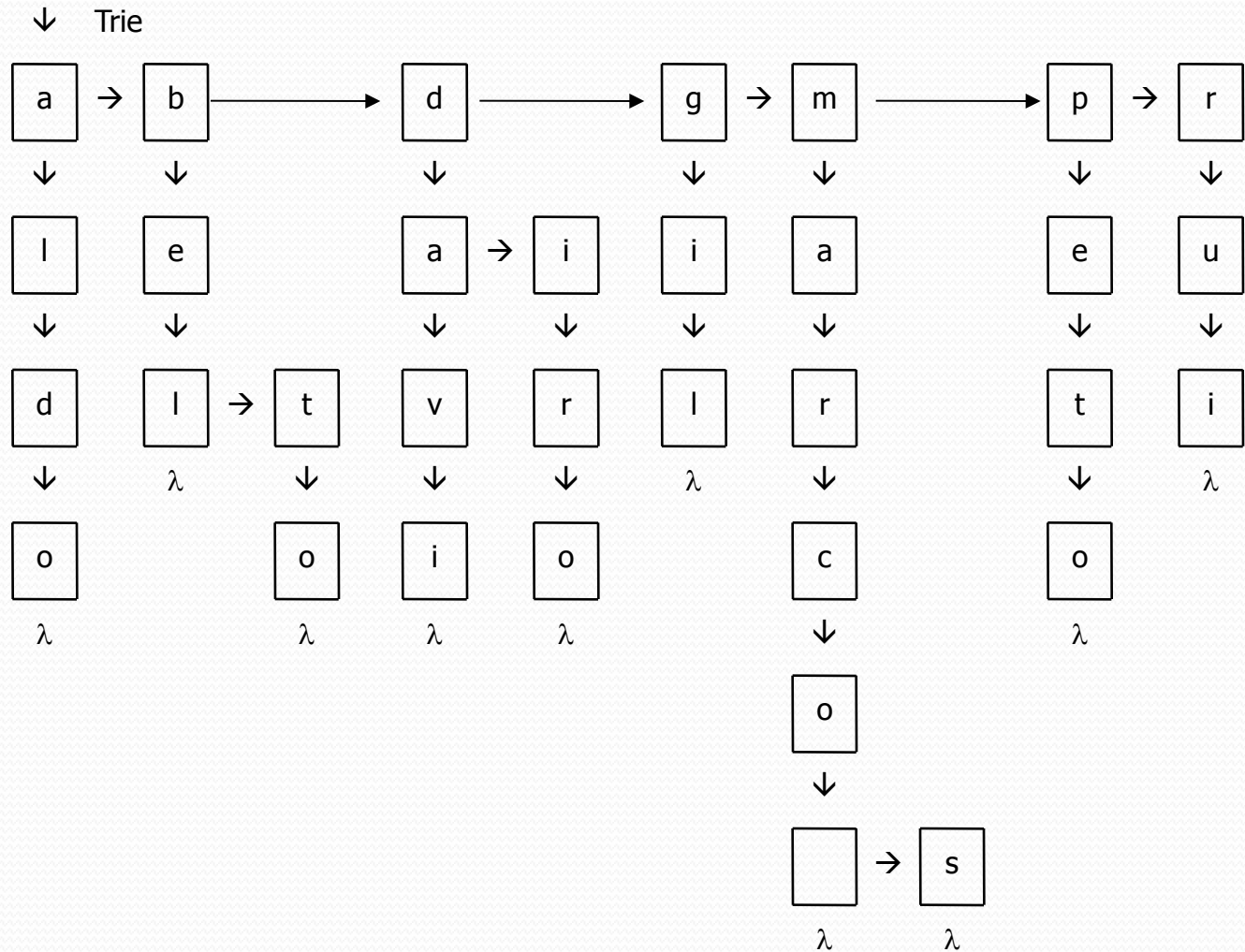


Construindo uma **trie dinâmica** (cada letra é mantida numa variável dinâmica) para armazenar:
davi, aldo,
beto, diro,
peto, rui, bel, gil,
marco,
marcos, dino.



Como, inserir
bel, gil, marco,
marcos e dino?

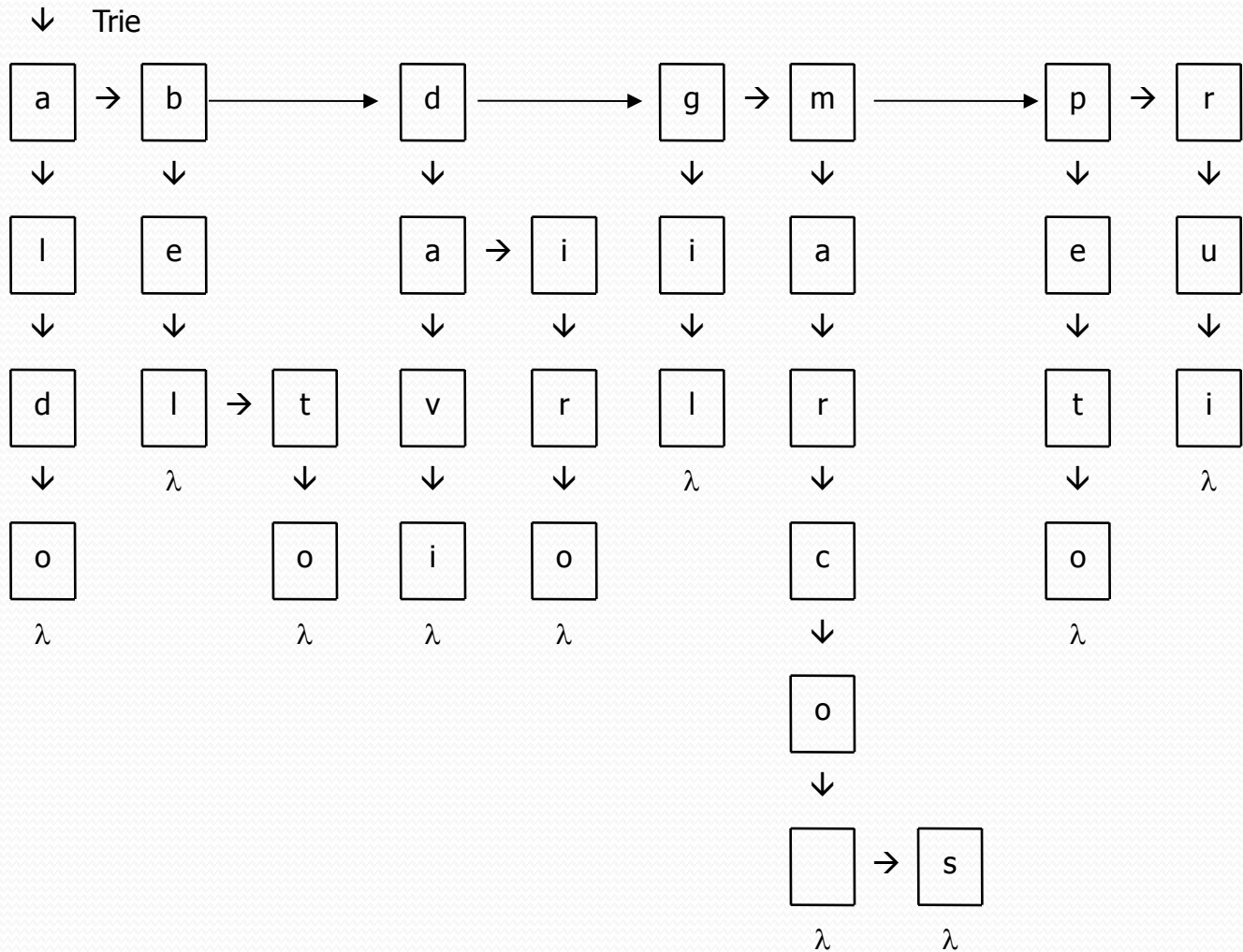
Construindo
uma **trie**
dinâmica
(cada letra é
mantida
numa variável
dinâmica)
para
armazenar:
davi, aldo,
beto, diro,
peto, rui, bel,
gil , marco,
marcos, dino.



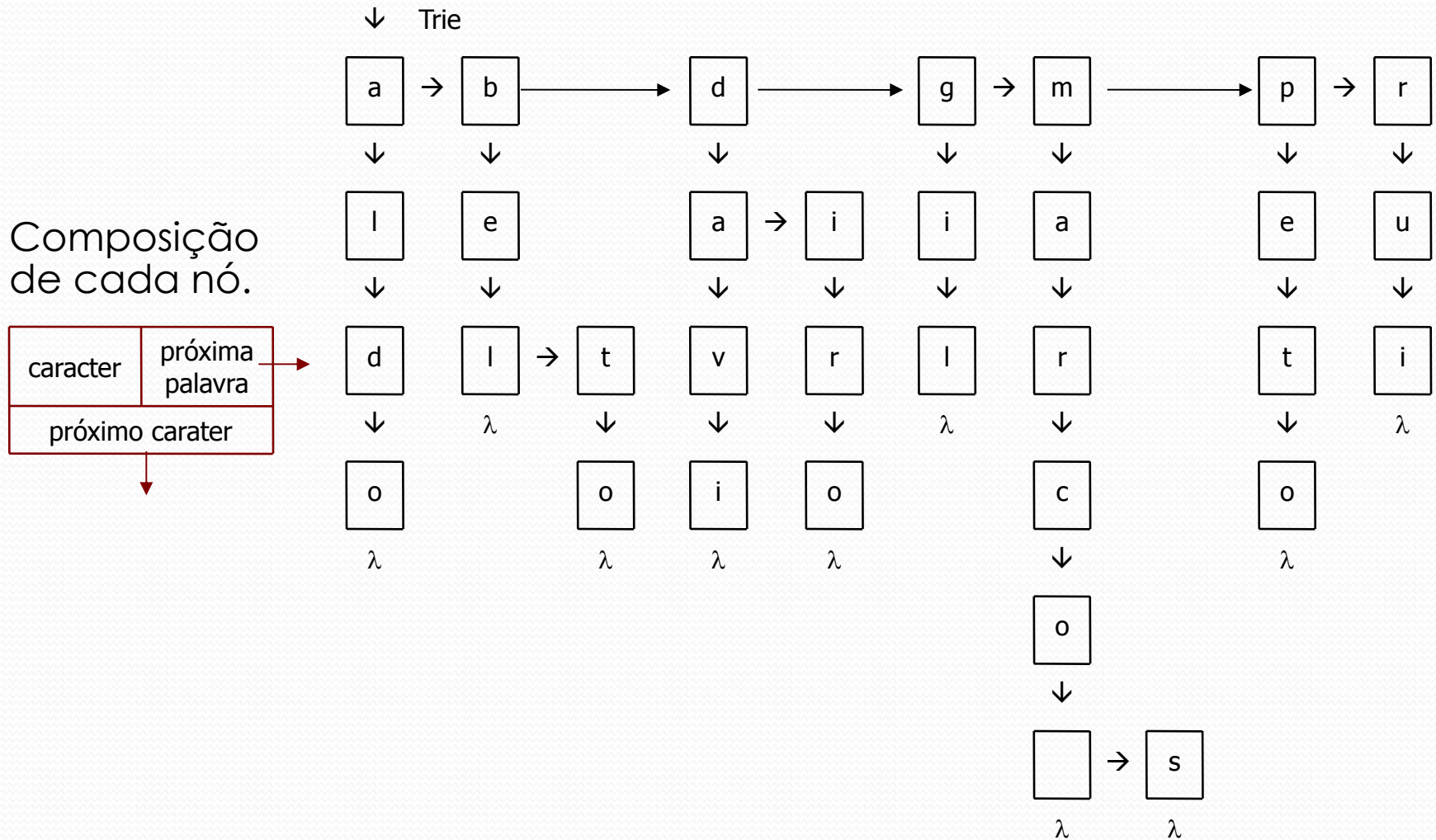
Trie Dinâmica



Como
“declarar”
cada nó?



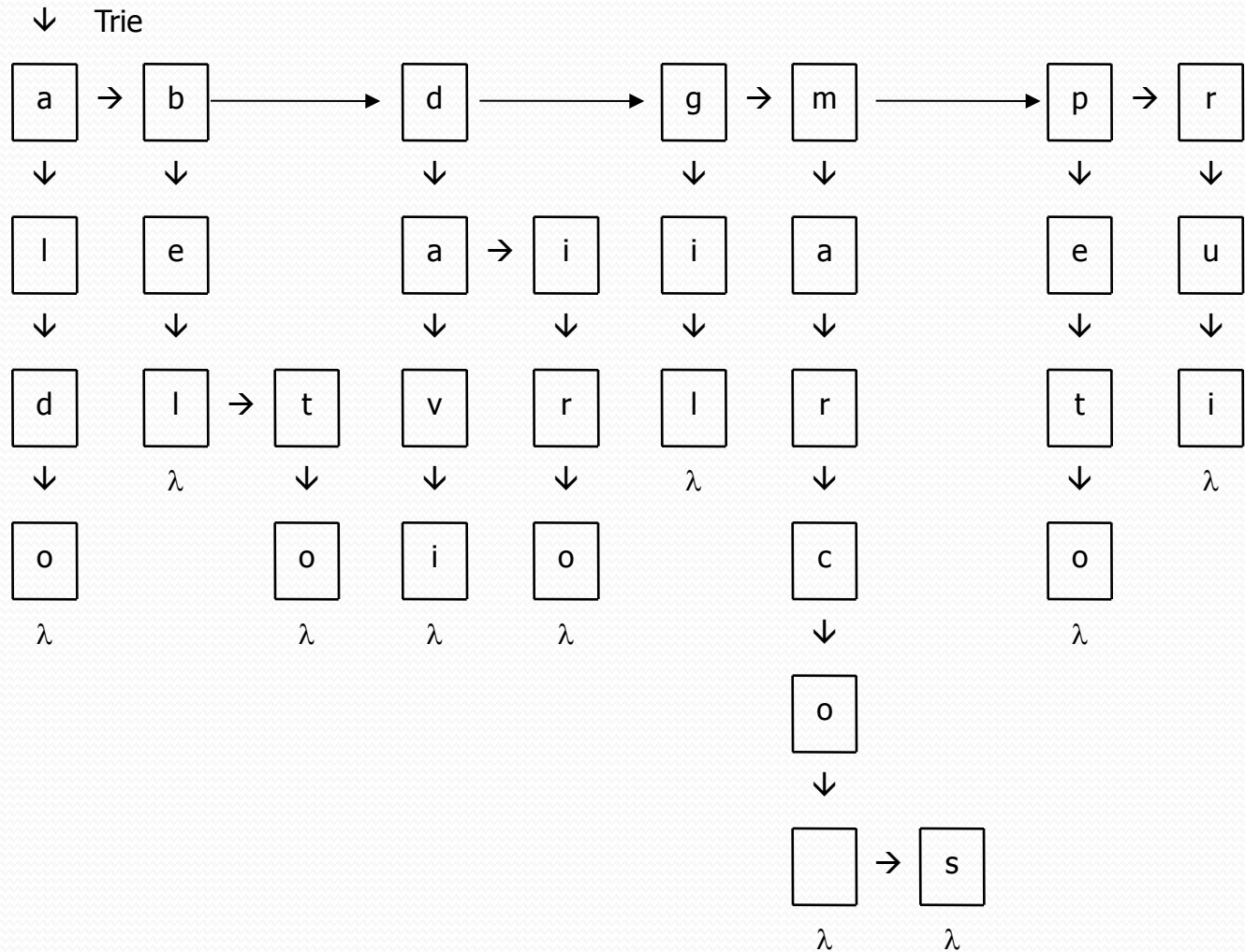
Trie Dinâmica



Trie Dinâmica



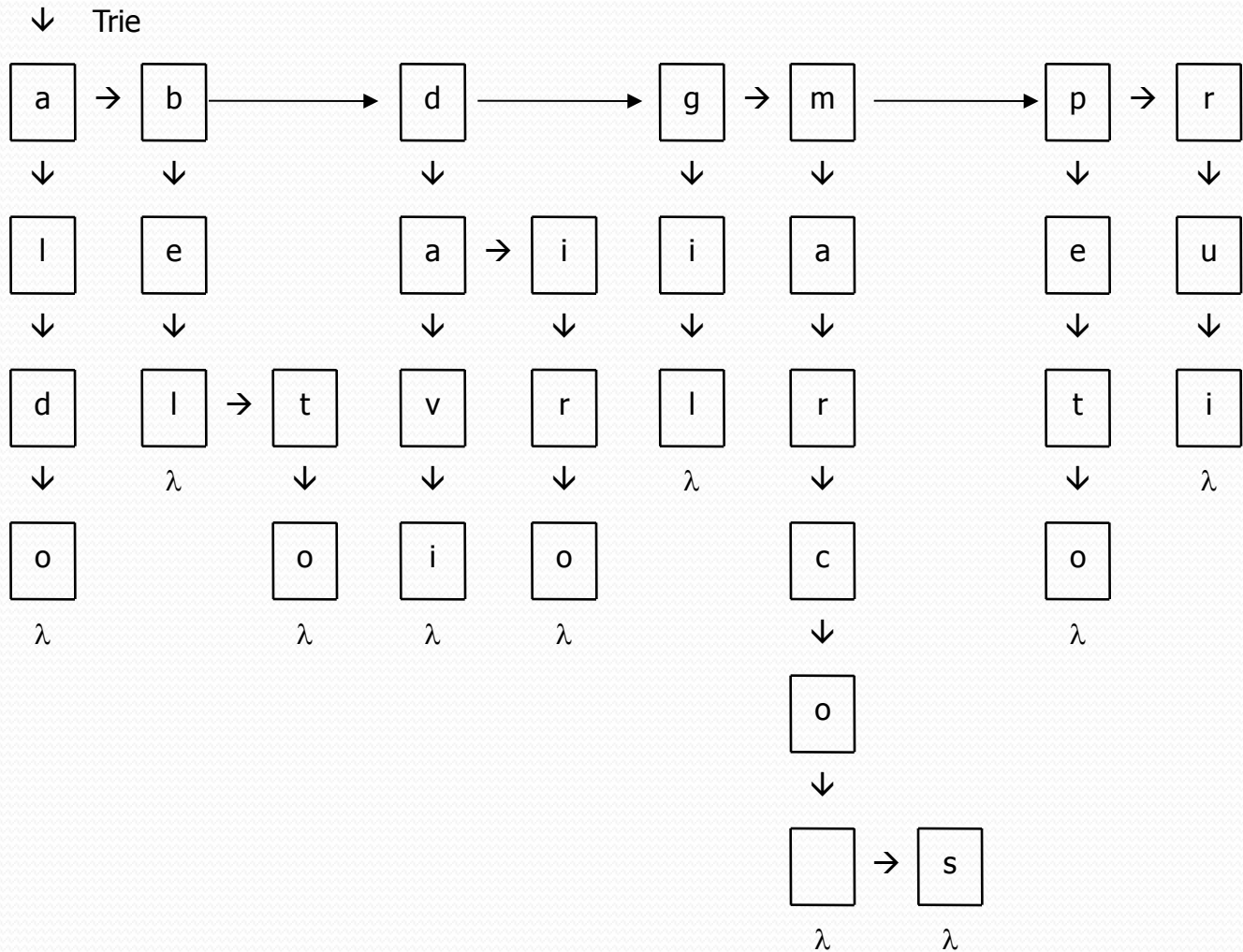
Como
efetuar
consulta
nesta?



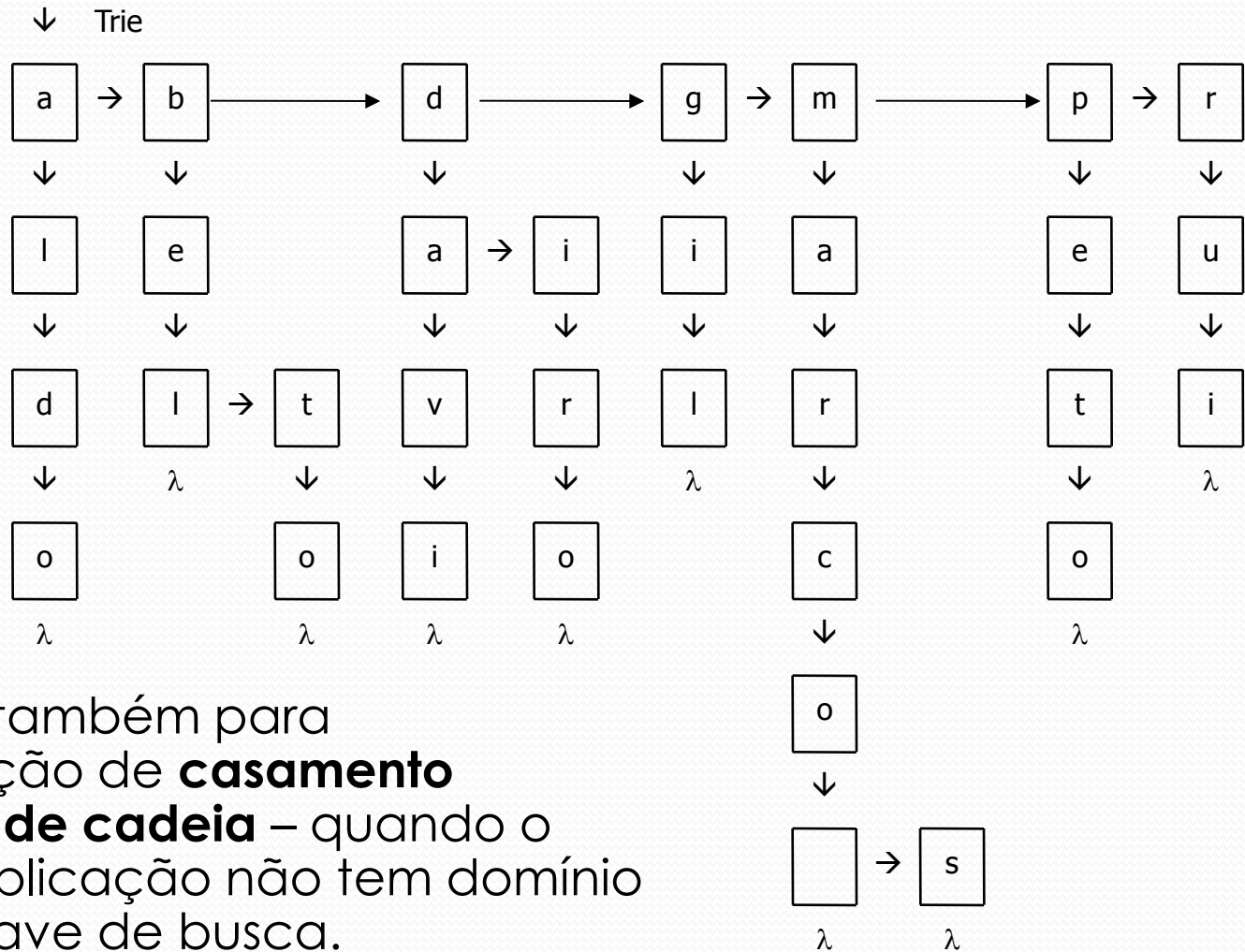
Trie Dinâmica



Como,
através desta,
implementar
dicionário?



Trie Dinâmica



Estrutura útil também para
implementação de **casamento
aproximado de cadeia** – quando o
usuário da aplicação não tem domínio
exato da chave de busca.



Exercício

Construir tries (estática e dinâmica) para manter: pinha, jaca, caju, coco, açaí, caja e pinhao.



Exercício

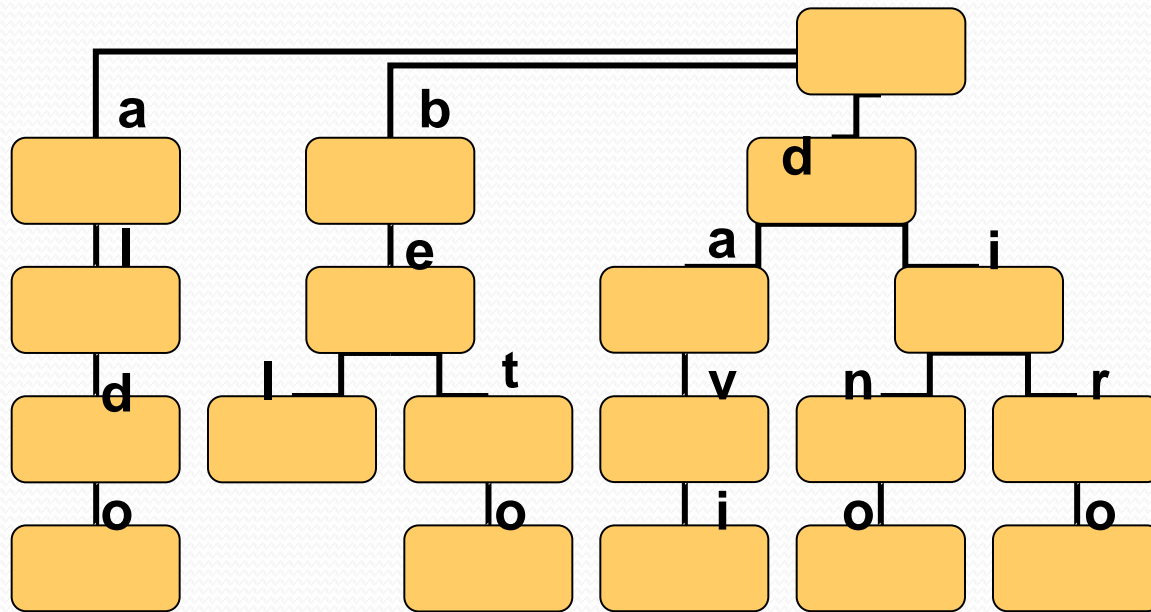
Que implementação da trie apresenta melhor desempenho: a estática ou a dinâmica? Justifique: (a) em velocidade de processamento, (b) em espaço de armazenamento



Exercício

Descrever a implementação do recurso auto-completar a partir de palavras mantidas em tries.

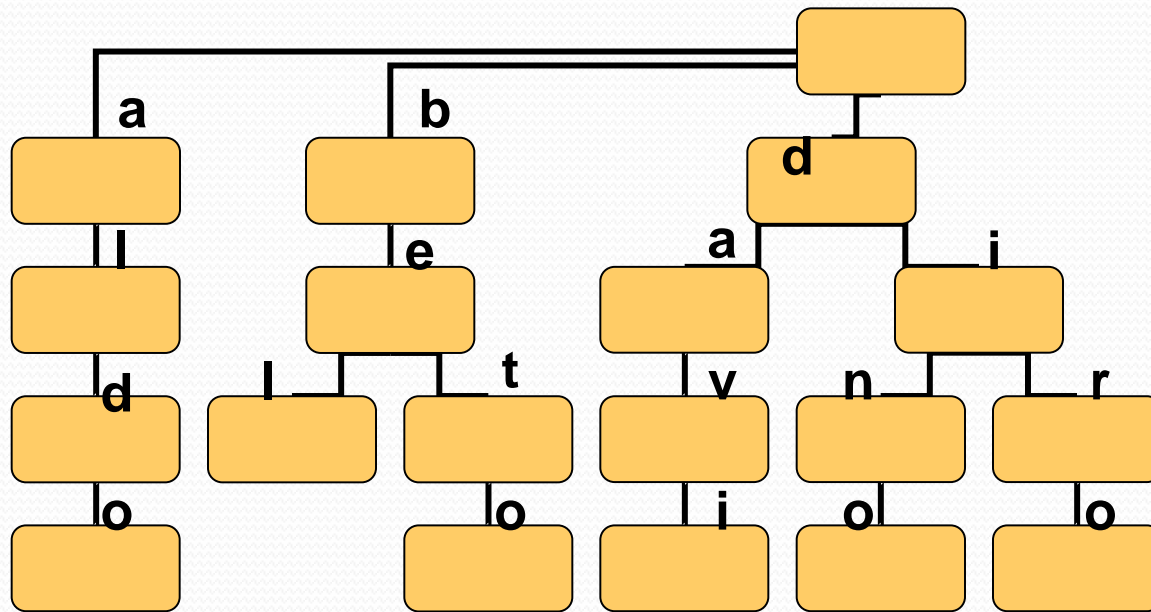
Árvore Digital de Pesquisa



Uma árvore digital para um alfabeto S com m caracteres (com itens ordenáveis) é uma árvore m -ária T , não vazia, tal que:

1. Se um nó qualquer x é o i -ésimo filho de seu pai, então x corresponde ao i -ésimo item do alfabeto S , sendo $1 \leq i \leq m$.

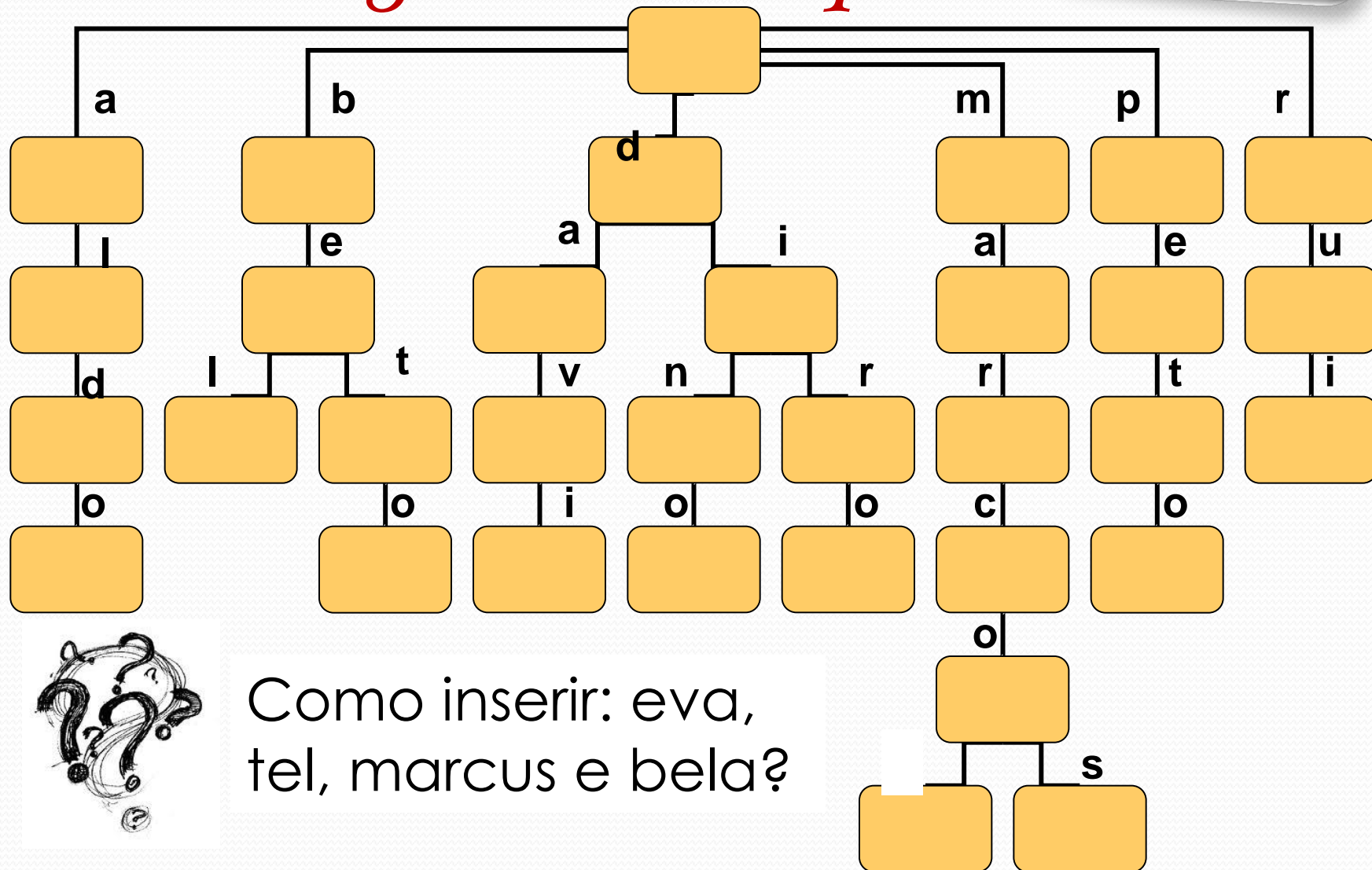
Árvore Digital de Pesquisa



Uma árvore digital para um alfabeto S com m caracteres (com itens ordenáveis) é uma árvore m -ária T , não vazia, tal que:

2. Para cada nó x , a sequência de dígitos definida pelo caminho desde a raiz de T até x corresponde no todo, ou em parte, a uma chave de pesquisa.

Árvore Digital de Pesquisa



Como inserir: eva,
tel, marcus e bela?



Como efetuar consulta nesta?





Exercício

Construir árvore digital para manter: pinha, jaca, caju, coco, açaí, caja e pinhao.



Exercício

Descrever a operação de consulta, a uma determinada palavra, numa árvore digital de pesquisa.



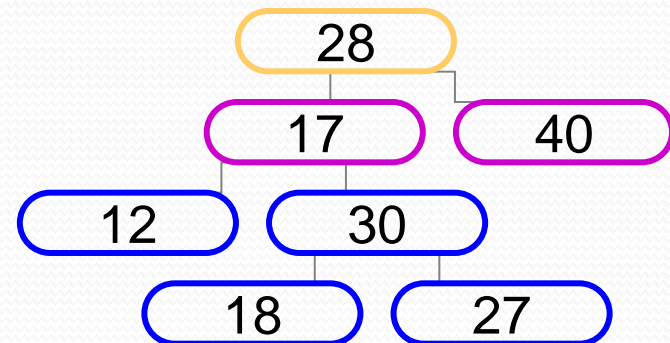
Exercício

Elaborar declaração de uma árvore digital de pesquisa, considerando o nosso alfabeto.

Árvore Digital de Pesquisa

Como estrutura de busca em texto, há ainda as **árvores digitais binárias** de pesquisa. Estas são compostas com base no código binário correspondente às chaves de pesquisas. Ou seja, corresponde a uma árvore digital de pesquisa cujo alfabeto apresenta somente dois itens.

Árvore digital de pesquisa com os dados: 28(011100)
17(010001) 30(011110)
27(011011) 40(101000)
12(001100) 18(010010)



Árvore Digital de Pesquisa



28

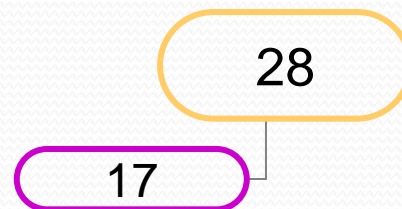
Construindo uma
árvore digital de
pesquisa com os
dados: 28(011100)
17(010001) 30(011110)
27(011011) 40(101000)
12(001100) 18(010010)

O primeiro dado inserido pode ser mantido na raiz da árvore. Mas há autores que consideram que a raiz da árvore não armazena dados, nem corresponde a qualquer dígito/dado.

Árvore Digital de Pesquisa



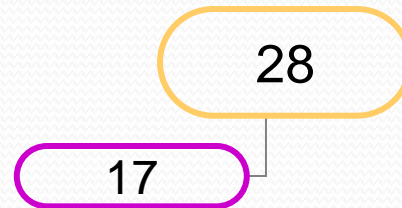
Construindo uma árvore digital de pesquisa com os dados: 28(011100)
17(010001) 30(011110)
27(011011) 40(101000)
12(001100) 18(010010)



A partir da segunda entrada, analisa-se a composição binária da chave, de primeiro ao último bit. Se o bit for 0 segue para esquerda da estrutura, se 1, pela direita.

Árvore Digital de Pesquisa

Construindo uma árvore digital de pesquisa com os dados: 28(011100)
17(010001) 30(011110)
27(011011) 40(101000)
12(001100) 18(010010)

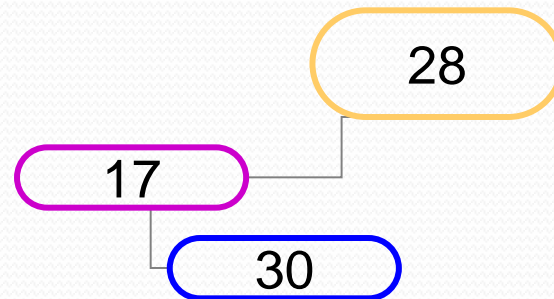


A partir da análise da composição binária da chave, esta pode ser armazenada no nó da estrutura, ou seja, adota-se o mesmo critério usado para composição das árvores digitais; onde as arestas definem a composição da chave.

Árvore Digital de Pesquisa



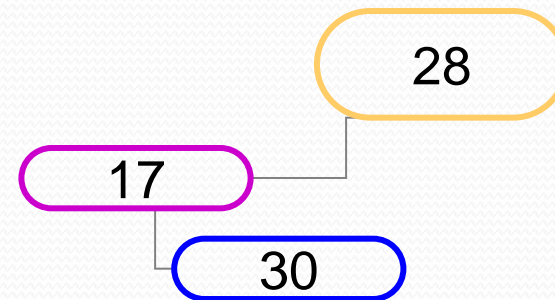
Construindo uma árvore digital de pesquisa com os dados: 28(011100)
17(010001) 30(011110)
27(011011) 40(101000)
12(001100) 18(010010)



Iniciando pela raiz, o primeiro bit de 30 é 0, deve-se seguir pela esquerda, a qual está ocupada. O segundo bit é 1, segue-se pela direita, como é encontrado aterramento, este é o ponto de inserção.

Árvore Digital de Pesquisa

Construindo uma árvore digital de pesquisa com os dados: 28(011100)
17(010001) 30(011110)
27(011011) 40(101000)
12(001100) 18(010010)



Como inserir os outros elementos na estrutura:

Árvore Digital de Pesquisa

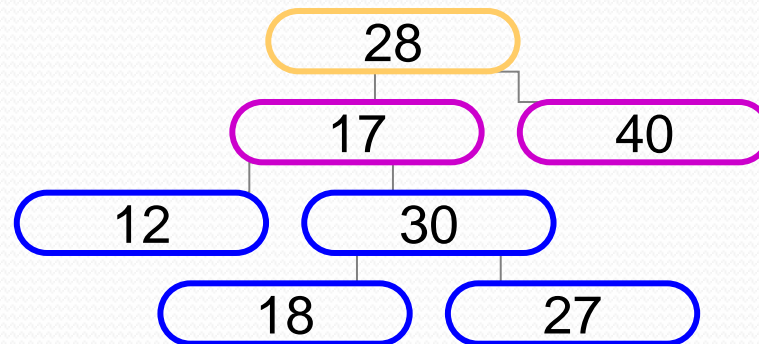
Construindo uma
árvore digital de
pesquisa com os

dados: 28(011100)

17(010001) 30(011110)

27(011011) 40(101000)

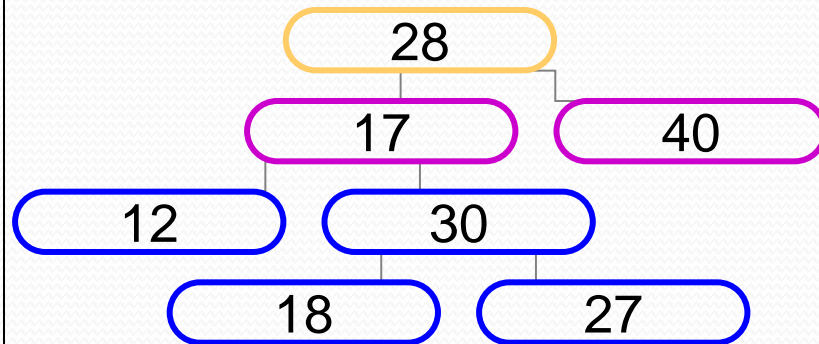
12(001100) 18(010010)



Como efetuar
consulta nesta?

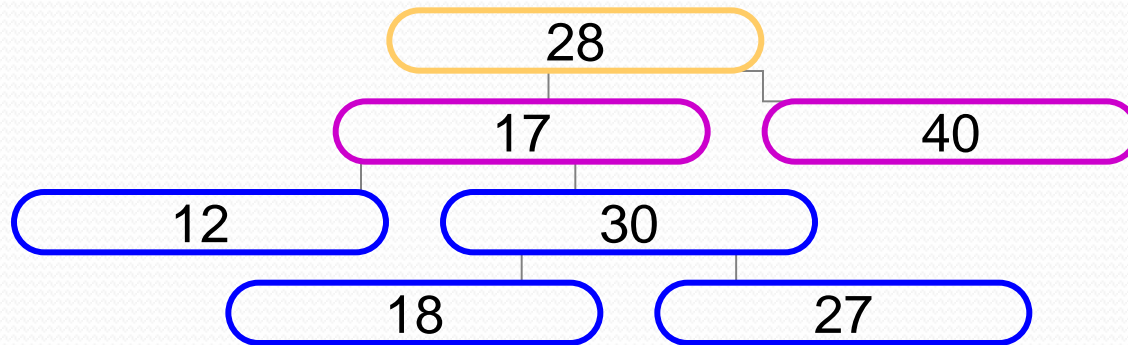
Árvore Digital de Pesquisa

Construindo uma
árvore digital de
pesquisa com os
dados: 28(011100)
17(010001) 30(011110)
27(011011) 40(101000)
12(001100) 18(010010)



A **consulta** é efetuada a partir do código binário correspondente à chave de pesquisa seguindo os mesmos critérios usados na construção da estrutura.

Árvore Digital de Pesquisa



Esta estrutura é passível de degeneração, como a maioria das árvores de busca.

Árvore Digital de Pesquisa

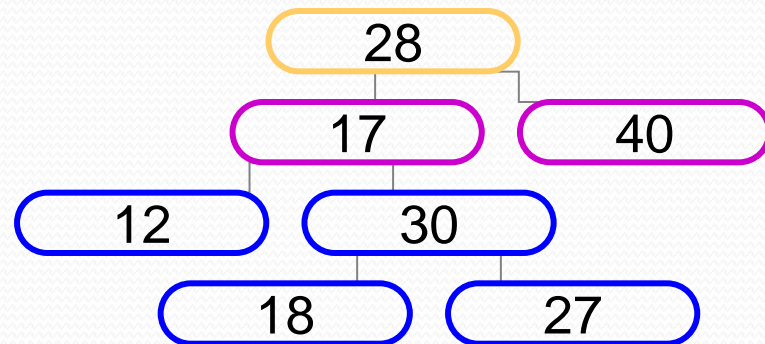
Construindo uma
árvore digital de
pesquisa com os

dados: 28(011100)

17(010001) 30(011110)

27(011011) 40(101000)

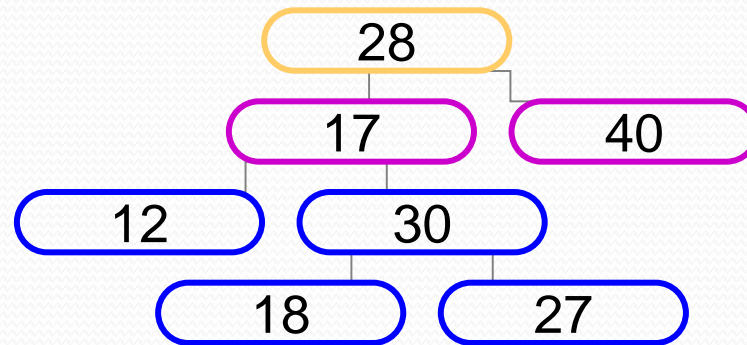
12(001100) 18(010010)



Como efetuar
exclusão nesta?

Árvore Digital de Pesquisa

Construindo uma
árvore digital de
pesquisa com os
dados: 28(011100)
17(010001) 30(011110)
27(011011) 40(101000)
12(001100) 18(010010)



Na exclusão qualquer nó pode ser substituído por seus sucessores; de preferência uma folha.

Árvore Digital de Pesquisa

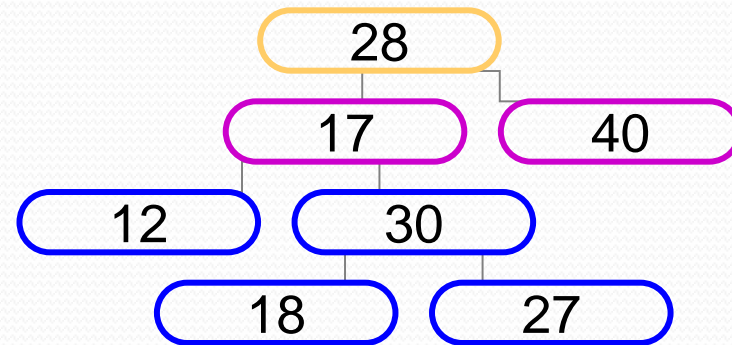
Construindo uma
árvore digital de
pesquisa com os

dados: 28(011100)

17(010001) 30(011110)

27(011011) 40(101000)

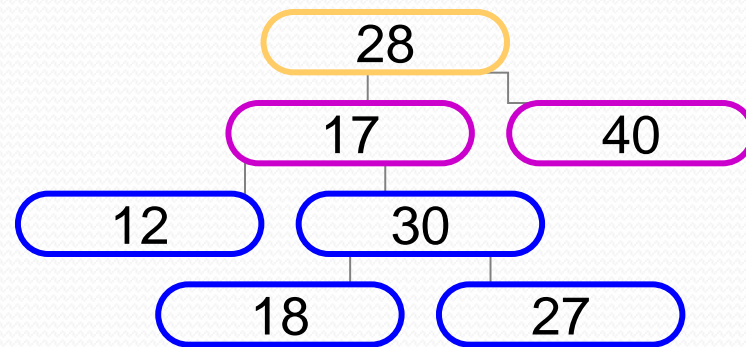
12(001100) 18(010010)



Onde aplicar esta?

Árvore Digital de Pesquisa

Construindo uma
árvore digital de
pesquisa com os
dados: 28(011100)
17(010001) 30(011110)
27(011011) 40(101000)
12(001100) 18(010010)

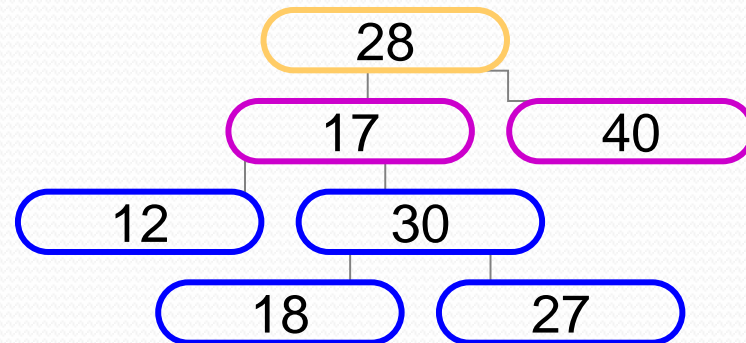


Podem ser aplicadas, por exemplo, em operação de baixo nível, para facilitar a comparação de símbolos.



Exercício

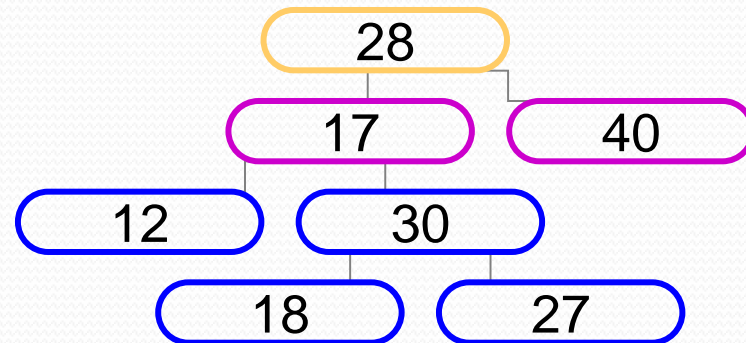
Partindo da árvore digital dada, efetue a inclusão das chaves 19, 50, 01 e 23.





Exercício

Partindo da árvore digital dada, efetue a exclusão da chave 30.





Exercício

Descreva os passos gerais da operação de consulta em árvores digitais de pesquisa.

Complementar Estudos...

File Organization and Processing

Allan L Tharp



Capítulo 11

Other Tree Structures
Tries & Digital Search Trees