

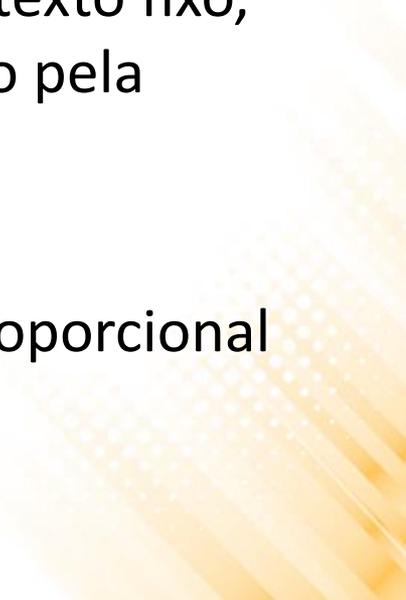
Projeto e Análise de Algoritmos

Aula 03 – Tries

Edirlei Soares de Lima
<edirlei@iprj.uerj.br>

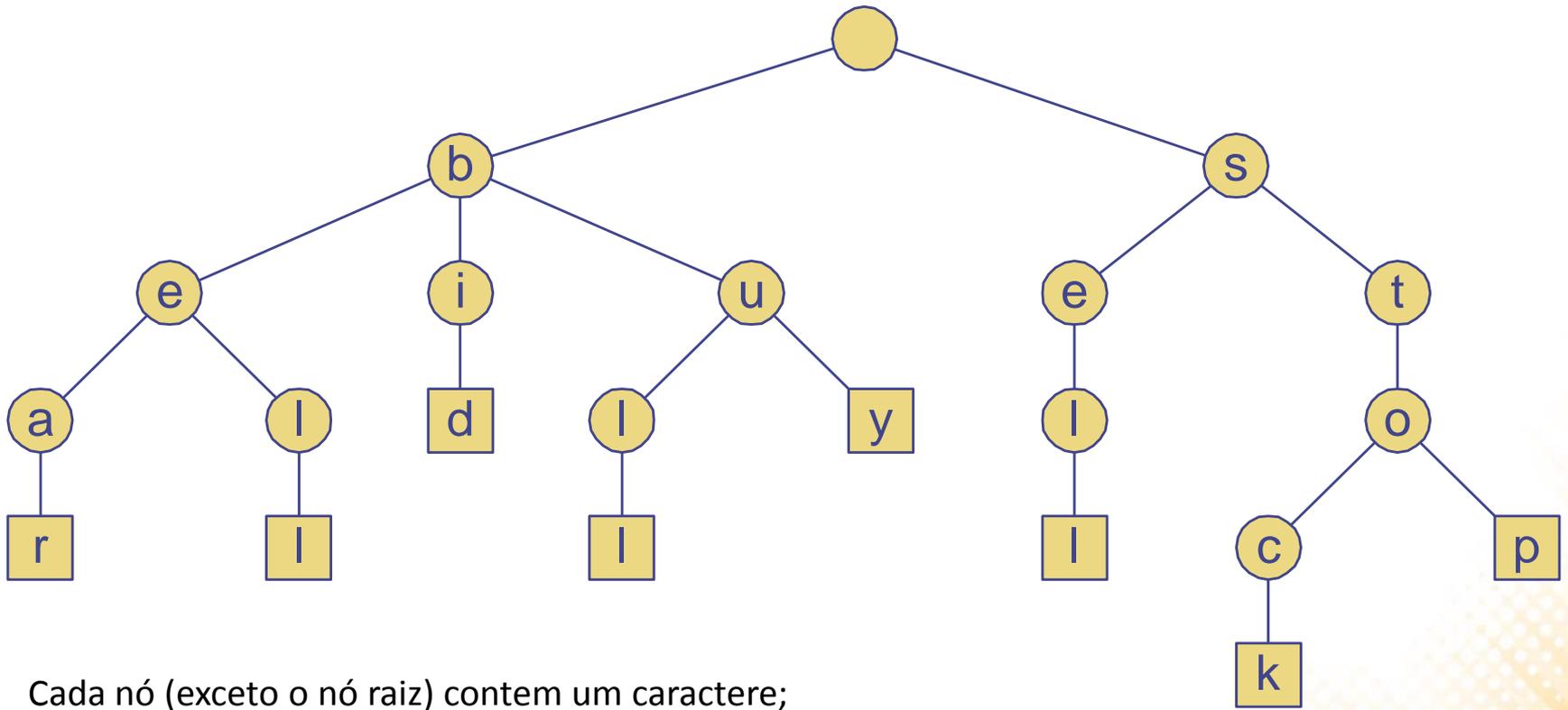


Tries

- Originado de "information re**TRIE**val" devido a aplicação em recuperação de informação.
 - Agilizam a busca através do pré-processamento do texto.
 - Quando uma série de consultas é realizada em um texto fixo, o custo inicial de pré-processar o texto compensado pela aceleração das consultas seguintes.
 - Permite a busca por padrões de texto em tempo proporcional ao tamanho do padrão buscado.
- 

Tries

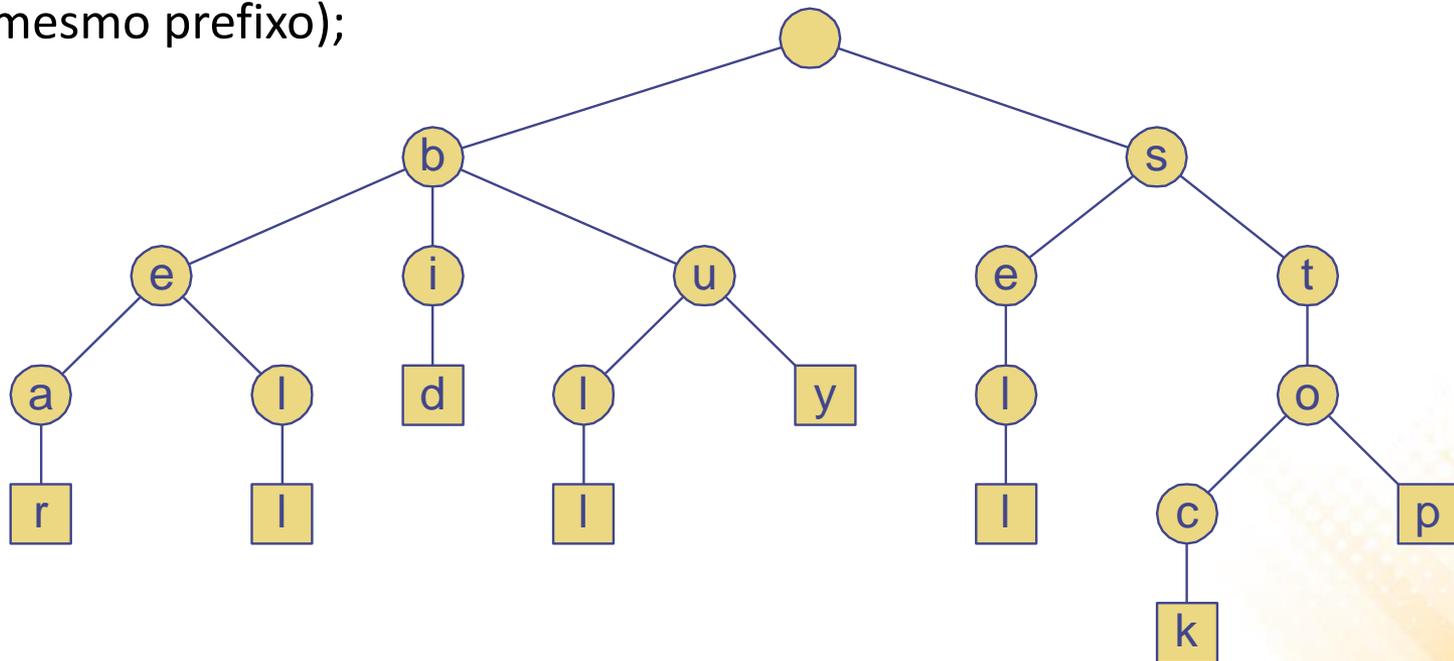
- $S = \{\text{bear, bell, bid, bull, buy, sell, stock, stop}\}$



- Cada nó (exceto o nó raiz) contém um caractere;
- Os nós filhos são ordenados alfabeticamente;
- Os caminhos entre o nó raiz e um nó folha contêm uma das strings $\in S$.

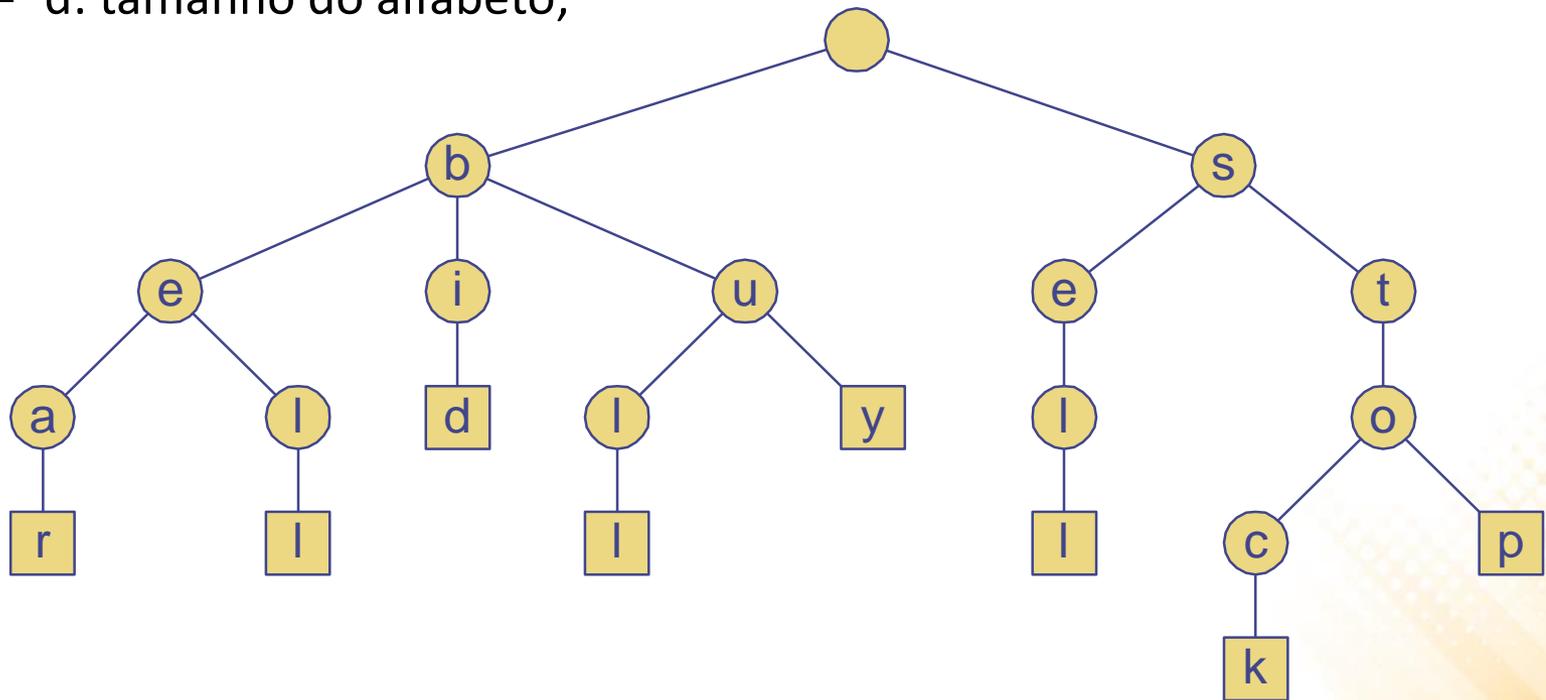
Tries - Análise

- Se o comprimento total do texto é n , o texto possui m palavras e d é o tamanho do alfabeto:
 - Cada nó possui no máximo d filhos;
 - A trie possui m folhas e tem altura igual ao comprimento da maior palavra;
 - O número de nós é $O(n)$ (n se nenhum par de palavras compartilha um mesmo prefixo);



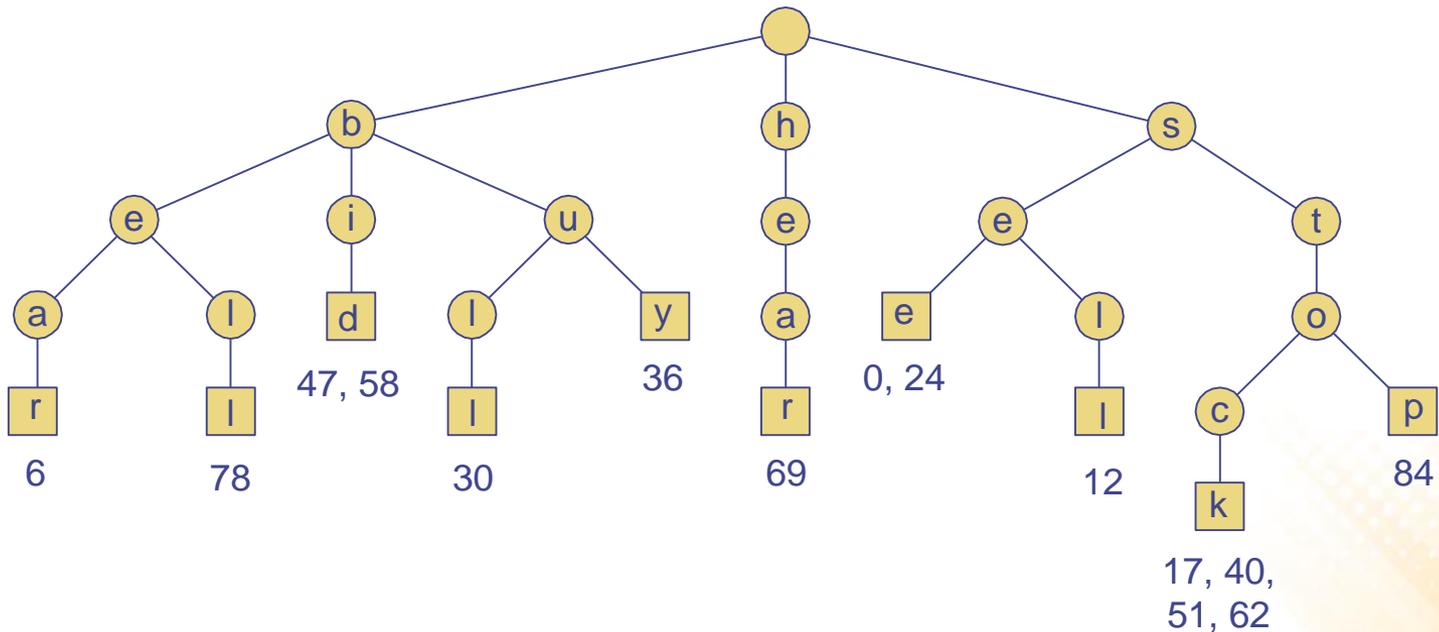
Tries - Análise

- Uma trie utiliza espaço $O(n)$ e permite realizar buscas, inserções e remoções em tempo $O(dm)$.
 - n : tamanho total das strings em S ;
 - m : tamanho da string buscada, inserida ou removida;
 - d : tamanho do alfabeto;



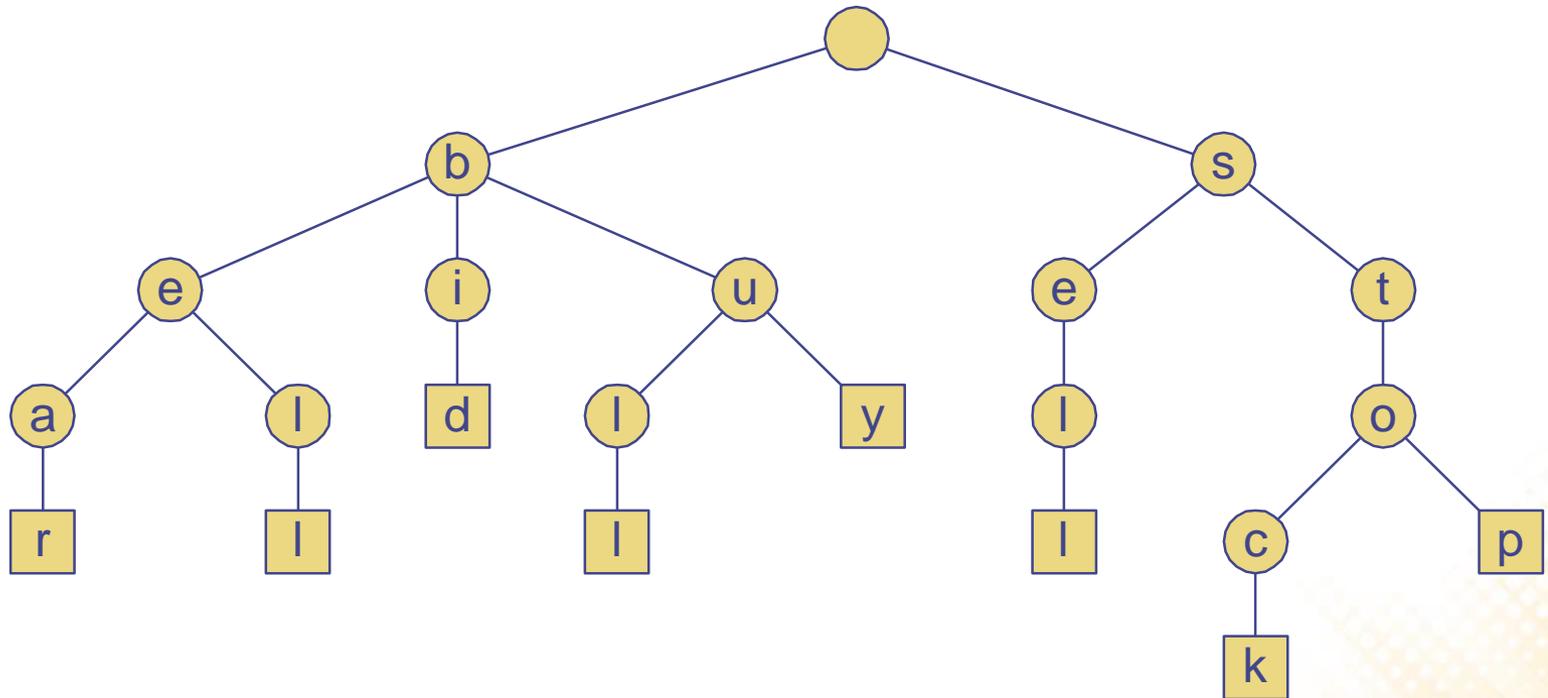
Tries – Encontrando Padrões

s	e	e		a		b	e	a	r	?		s	e	l	l		s	t	o	c	k	!	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
s	e	e		a		b	u	l	l	?		b	u	y		s	t	o	c	k	!		
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	
b	i	d		s	t	o	c	k	!		b	i	d		s	t	o	c	k	!			
47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68		
h	e	a	r		t	h	e		b	e	l	l	?		s	t	o	p	!				
69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88				



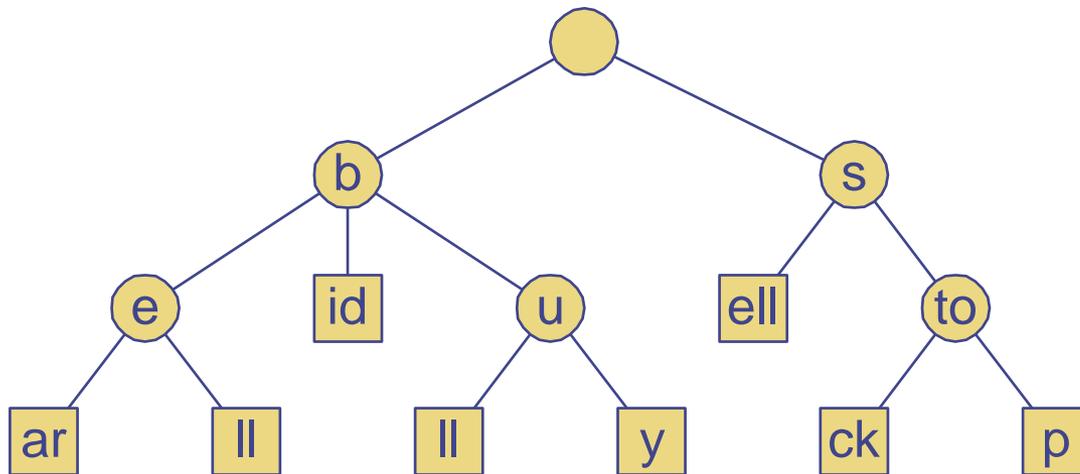
Tries Comprimidas

- Em uma trie comprimida:
 - Cada nó interno deve ter pelo menos 2 filhos. Nós que contem apenas 1 filho são concatenados, formando um novo nó;



Tries Comprimidas

- Em uma trie comprimida:
 - Cada nó interno deve ter pelo menos 2 filhos. Nós que contem apenas 1 filho são concatenados, formando um novo nó;



- Menor número de nós, porém a informação de cada nó ocupa mais espaço.
- Solução?

Tries Comprimidas

- Representação compacta para uma trie comprimida:
 - Armazena intervalos de índices nos nós em vez de substrings;
 - Utiliza espaço $O(s)$;

S[0] =

0	1	2	3	4
s	e	e		

S[4] =

0	1	2	3
b	u	l	l

S[7] =

0	1	2	3
h	e	a	r

S[1] =

0	1	2	3
b	e	a	r

S[5] =

0	1	2
b	u	y

S[8] =

0	1	2	3
b	e	l	l

S[2] =

0	1	2	3
s	e	l	l

S[6] =

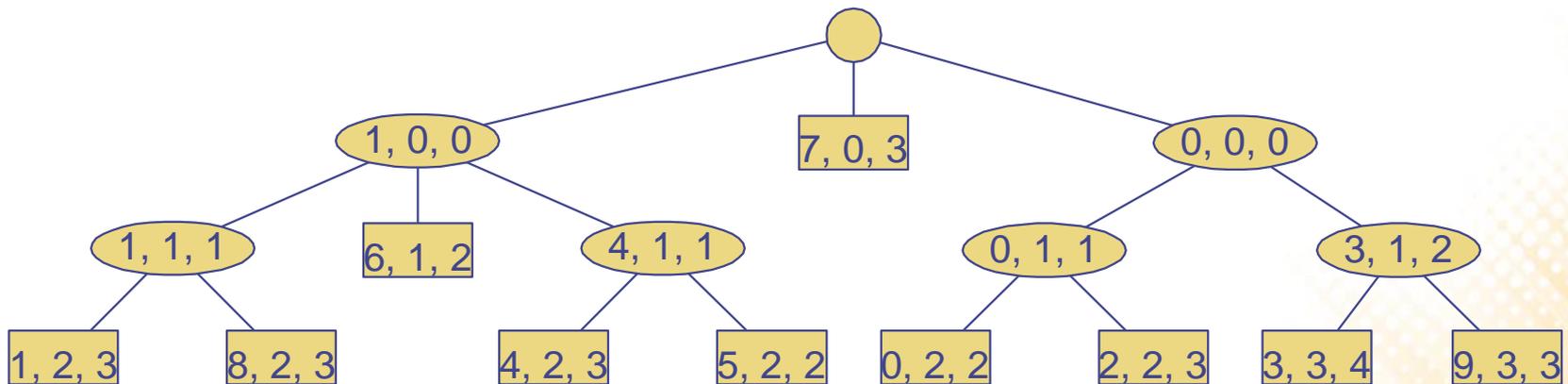
0	1	2
b	i	d

S[9] =

0	1	2	3
s	t	o	p

S[3] =

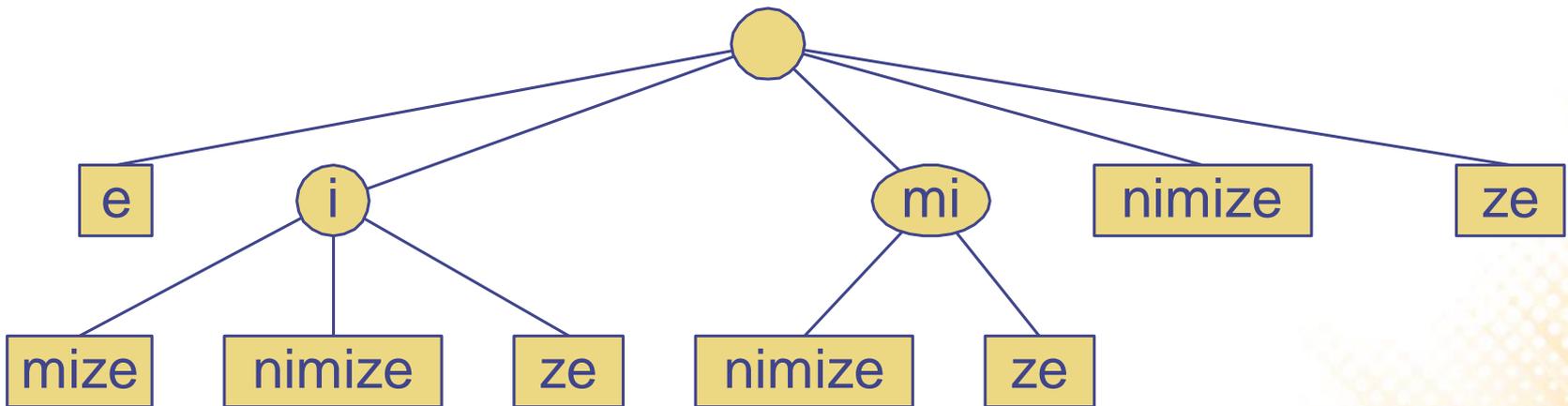
0	1	2	3	4	5	6
s	t	o	c	k		



Suffix Trie

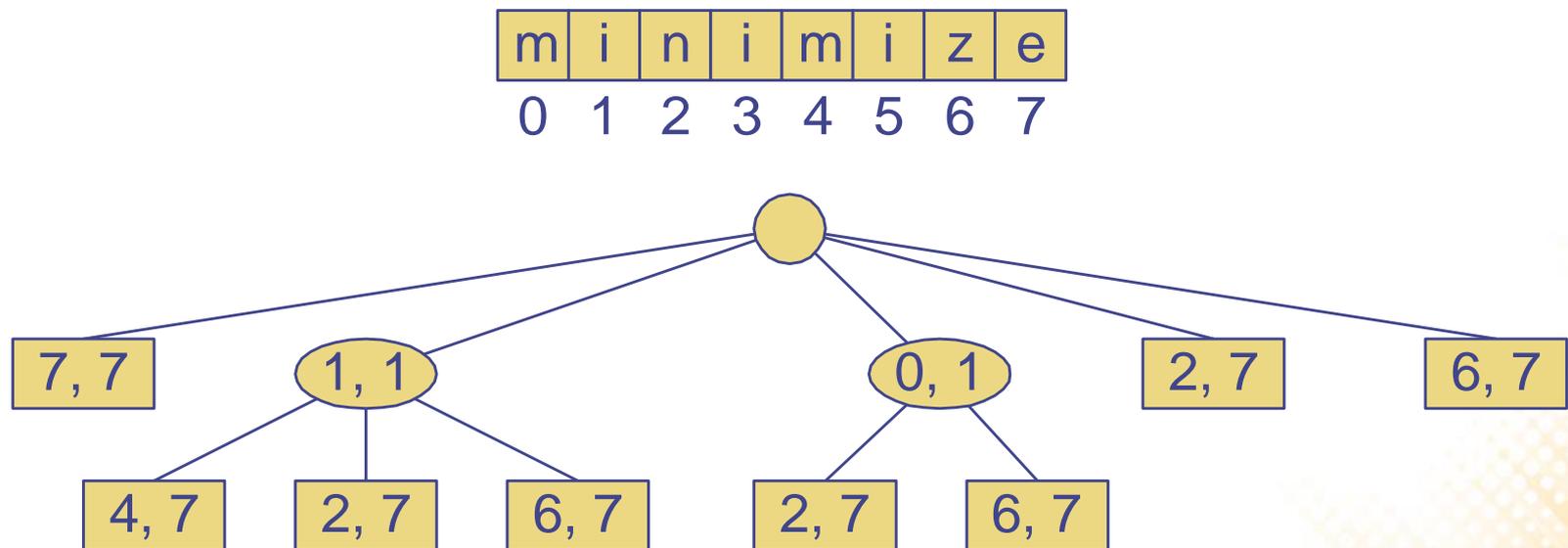
- Uma suffix trie de uma string X é uma trie comprimida de todos os sufixos de X .

m	i	n	i	m	i	z	e
0	1	2	3	4	5	6	7



Suffix Trie – Análise

- Representação compacta da suffix trie de uma string X de tamanho n usando um alfabeto de tamanho d .
 - Permite realizar buscas por padrões arbitrários em $O(dm)$, onde m é o tamanho do padrão a ser buscado.



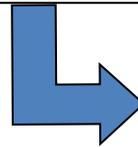
Tries – Aplicações

- Motores de busca armazenam o seu índice de busca (coleção de palavras que podem ser buscadas) em uma trie comprimida.
 - Cada nó folha da trie é associada a uma palavra e a uma lista de páginas web (URLs) contendo a palavra;
 - A trie é mantida em memória;
 - A lista de ocorrências é mantida em memória externa e pode ser ranqueada por relevância;
 - Buscas booleanas envolvendo conjuntos de palavras (exemplo: *Análise and Algoritmos*), correspondem a conjuntos de operações (exemplo: intercessão) nas listas de ocorrências;
 - Técnicas extras também podem ser incluídas (exemplo: eliminação de stopwords).

Tries – Aplicações

Base de Documentos

Documento	Texto
1	Pease porridge hot, pease porridge cold
2	Pease porridge in the pot
3	Nine days cold
4	Some like it hot, some like it cold
5	Some like it in the pot
6	Nine days old



Lista de ocorrências

No	Termo	(Docs; Pos)
1	cold	(1;6), (4;8)
2	days	(3;2), (6;2)
3	hot	(1;3), (4;4)
4	in	(2;3), (5;4)
5	it	(4;3,7), (5;3)
6	like	(4;2,6), (5;2)
7	nine	(3;1), (6;1)
8	old	(3;3), (6;3)
9	pease	(1;1,4), (2;1)
10	porridge	(1;2,5), (2;2)
11	pot	(2;5), (5;6)
12	some	(4;1,5), (5;1)
13	the	(2;4), (5;5)

Vocabulário

Ocorrências e posições

Tries – Exercícios

- 1) Escreva um algoritmo para encontrar padrões em uma trie.
 - 2) É possível tornar esse algoritmo mais eficiente?
- 